**2-1 Journal: What Makes a Productive Code Review?**

**Part 1:**Code review is the process of examining and evaluating another developer’s code to identify errors, improve the quality of the code, and ensure it adheres to best practices. It’s a collaborative process where developers work together to catch potential issues, improve maintainability, and share knowledge.

Code review is essential for computer science professionals because it helps maintain high quality standards in software development. It’s not just about catching bugs, it’s also about teamwork, sharing knowledge, and ensuring the codebase remains understandable for everyone. Reviewing code regularly can help prevent future problems and encourage developers to write cleaner and more efficient code.

Some best practices I read about include reviewing small portions of code at a time to avoid getting overwhelmed, using a checklist to standardize what to look for, and providing constructive feedback rather than just pointing out mistakes. Code reviews should happen after the developer has thoroughly tested their code but before it’s merged into the main branch. This timing ensures the code is ready for review and avoids wasting time on incomplete work.

**Part 2:**

For recording my code review, I plan to use OBS Studio. It’s a tool I’m already comfortable with, and it allows me to create clear video walkthroughs of my code.

My approach to creating a code review script is straightforward. For each of the three categories; software engineering and design, algorithms and data structures, and databases. I’ll start by explaining the purpose of the code and its functionality. Then, I’ll identify areas for improvement based on the rubric and code review checklist. I’ll focus on things like structure, readability, and performance, making sure I provide specific examples and suggestions. This way, the review is organized and easy to follow.