Individual Reflection Paper for

Real-time sys-monitor Group Project

1. Personal Contributions

My primary role in this project was to implement core system monitoring functionalities and design the overall architecture, including data models and ensuring robust data handling. My contributions included setting up the project structure, creating modules for data retrieval (CPU, memory, disk, network statistics), and implementing file logging for historical analysis.

2. Technical Challenges

A key technical challenge was handling real-time data without causing excessive system load. To address this, I optimized the data retrieval and update intervals to balance real-time performance with system efficiency. Additionally, managing asynchronous tasks in Rust was challenging, particularly when integrating with the Iced framework, which was new to me. After research and experimentation, I implemented a solution using Rust's async capabilities, which helped me understand task scheduling and concurrency better.

3. Rust and Iced Framework

Using Rust was a productive experience, particularly due to its memory safety and performance benefits in system programming. The language's strict ownership and borrowing model was challenging at first, but it ultimately helped in building a secure application. Working with the Iced GUI framework allowed me to explore Rust's capabilities in frontend development. Iced’s declarative approach simplified GUI management, although it had limited documentation. I supplemented my learning with community resources, such as this tutorial on GitHub.

4. Teamwork and Collaboration

Working in a team allowed us to divide responsibilities according to our strengths, which fostered productivity. However, syncing code and managing dependencies in Rust sometimes led to merge conflicts. To mitigate this, we implemented a regular pull request review system, ensuring everyone stayed updated on changes and reduced integration issues. Open communication was key to navigating these collaboration challenges.

5. Learning Outcomes

This project significantly enhanced my skills in Rust, especially in concurrency and real-time data processing. I gained hands-on experience with the Iced framework, understanding its strengths and limitations in GUI development. The project also improved my collaboration and communication skills.

6. Project Retrospective

If I were to redo this project, I would focus on better initial planning, particularly around module integration, to reduce rework. I would also separate the code for GUI and data retrieval within ‘`system_monitor.rs`’ for clearer error handling and improved modularity. Additionally, I would explore alternative GUI frameworks that support charts and graphs, as our current project presents only textual data for real-time system metrics.