

Name: Harshil Patel

Course: CS5002

Task: Assignment 6 – Self Assessment

Part A:

Throughout the senior design project, my primary responsibility was the development and integration of the web-based monitoring interface for our Windows Process Protector system. I designed and implemented a dynamic, responsive dashboard using modern web technologies including HTML5, CSS3, and React, ensuring real-time display of process information, threat alerts, and system logs. I also led efforts to integrate the backend security modules, developed in C++, with the frontend via secure communication protocols. As our project progressed into the spring semester, I refined UI elements to reflect severity levels, added historical log viewing and export functionality, and supported interactive process controls. I applied my previous experience with frontend frameworks and learned to adapt those skills to a cybersecurity-focused system with real-time communication requirements.

I also contributed extensively to testing, documentation, and presentations. I developed and simulated test cases for DLL injection, remote thread creation, and abnormal memory access, ensuring that our UI reflected alerts correctly and intuitively. My work required close collaboration with Bartosz to synchronize UI features with backend modules, especially as we added behavior analysis and network monitoring subsystems. One of my biggest challenges was designing a user interface that stayed lightweight and stable while processing real-time data streams. Overcoming these challenges helped me build stronger full-stack development skills, improve my debugging abilities, and understand secure data handling in cross-platform systems. I'm proud of how the final UI turned out—visually clean, technically sound, and functionally robust.

Part B:

As a group, we successfully designed and built a comprehensive, modular process protection system that integrated kernel-level monitoring, code integrity verification, behavior analysis, and a real-time web interface. Our system detects various cheating behaviors, flags unauthorized process modifications, and displays threat data live in a user-accessible dashboard. From idea to implementation, we translated research into practice while balancing system performance and security. One of our greatest team achievements was developing an architecture where frontend and backend operated asynchronously but remained fully in sync, thanks to clear communication protocols and well-defined APIs.

Working in a team taught me the value of accountability, consistency, and iterative feedback. Bartosz and I maintained a strong, respectful dynamic where we played to our strengths—he led kernel-level development and I drove frontend integration—but we frequently cross-collaborated to debug, test, and document features. We had regular meetings, always shared progress through GitHub, and made key design decisions together. My teammate, Bartosz, also deserves special recognition whose dedication to understanding low-level system behavior and ability to write clean, functional C++ code laid the technical foundation for everything we built. Compared to my teammate, I believe our efforts were balanced and complementary. This project proved how productive a small, focused team can be when communication is strong and responsibilities are clear.