Course Name: Computing with IT Management

Year/Group: 4th Year ITM

Semester: 8

Module Title: IT Management Finals Project

Lecturer Name: Enda Lee, David White, Fernando Perez Tellez

GAME PERFORMANCE AND IT MONITORING DASHBOARD

Student Name: Ricardo Danganan Jnr

Student Email: x00191395@mytudublin

Submission Date: 04/02/2025

Due Date: 04/02/2025

Table of Contents

Purpose of this Project....2
Problem Statement....2
Project Goals....2
Project Objectives....2
Core Features....2
Functionalities Delivered inPhase 1....2
Planned Enhancements....2
Technology Used....3
Frontend:....3
Backend:....3
APIs & Tools:....3
Development Phases of This Project....3
Phase 1: System Monitoring (Ongoing)....3
Phase 2: Latency Monitoring and Network Metrics (Upcoming)....4
Phase 3: Game Optimization Module....4
Phase 4: Historical Data Analysis and Trends....4
Risk Assessment....5
Identified Risks & Mitigation Strategies:....5
Conclusion....5

Purpose of this Project

Problem Statement

Performance bottlenecks in the confines of video games such as, FPS drops, heating, high CPU/GPU usage and latency all contribute detrimentally to gaming experience.

Multiplayer gamers also run into latency and network stability issues that impact competitive play. Players needs a performance Dashboard with real-time monitoring and good recommendations for optimization that improves their gaming experience.

Project Goals

This project focuses on creating a dashboard application for Windows that tracks gaming performance metrics in real time and presents them in a visually appealing manner. The dashboard will have:

- The monitoring includes CPU/GPU usage, temperatures, and RAM usage, disk activity, and latency for multiplayer gaming.
- Analyse the provided information for optimization recommendations.
- Utilize Steam API to identify installed games and adjust performance settings accordingly.
- Provide visual and audio cues for critical performance limit thresholds.
- Keep historical performance data for trend analysis.
- Scalable for future expansion with multi-system monitoring, cloud integration, etc.

Project Objectives

- Real-Time Monitoring Including FPS, CPU/GPU usage, temperatures, RAM and disk activity.
- Latency Monitoring that Monitors ping and network traffic for online gaming
- Optimization Recommendation that Provides system settings to increase gaming performance.
- User Alerts that use Visual and audio notifications in cases where performance thresholds are exceeded.
- Advanced Competitive Analysis that Keeps track of trends with log and graphs for more profound performance understanding
- Steam API Integration that Determines what games are installed for benchmarking and optimizations.

Core Features

Functionalities Delivered in Phase 1

Monitor Performance in Real-Time that Monitors CPU, GPU, RAM, and disk activity, along with simulated CPU temperature.

- All Systems Alert Provides both visual and auditory alerts when critical thresholds are exceeded.
- Interactive Dashboard that Displays live graphs with the help of React js and Chart. js.
- Back-end Data Handling Node used to pull system information effectively, also utilises js and PowerShell scripts.

Planned Enhancements

- Latency Monitoring so you could now measure ping and network bandwidth for multiplayer gaming

- Multi-Platform Support that Modifies scripts for Linux-based systems.
- Optimization that Detects games to adjust settings automatically.
- Cloud monitoring that Enables remote access to performance data.
- Multi-System Monitoring that Lets you monitor multiple devices in LAN environment.
- In-Game Overlay that Shows performance stats inside games.
- JSON data from the scripts that's going to be stored in database.

Technology Used

Frontend:

- React. js to create the interactive part of the UI.
- Chart. js for displaying the real-time performance graphs.
- CSS & Bootstrap For responsive design and styling.

Backend:

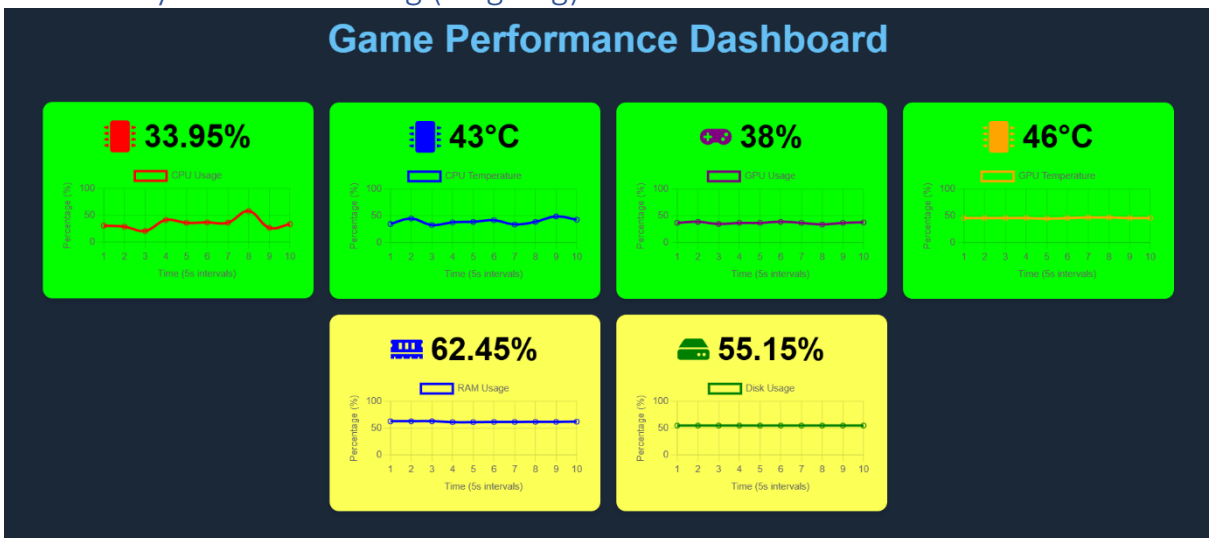
- Node. js Data request and monitoring in real-time.
- Express.js API framework for managing system metric requests.
- WebSocket that would allow real-time updates of CPU, GPU, RAM, and latency metrics without constantly refreshing the page
- Windows PowerShell Scripts that Retrieve metrics such as CPU/GPU Usage and Temperature Related System Information

APIs & Tools:

- Steam API that Retrieves the list of installed games when making recommendations for optimization.
- (Possible Feature) NVIDIA API for future integration (AI-driven optimization and DLSS settings)

Development Phases of This Project

Phase 1: System Monitoring (Ongoing)



Summary: This phase focused on building the foundation of the dashboard, implementing real-time monitoring for CPU, GPU, RAM, and disk usage, alongside alert notifications. The data retrieval process was structured using PowerShell scripts, and the frontend was built using React.js and Chart.js.

Possible Improvements:

- Enhance CPU temperature retrieval without requiring admin privileges.
- Optimize backend scripts for better performance and reduced processing load.
- Improve user customization for alert thresholds and notifications.
- Implement a cloud-based synchronization system to allow users to access performance data remotely.

Phase 2: Latency Monitoring and Network Metrics (Upcoming)

Summary: This phase will introduce network performance monitoring, including measuring ping, network bandwidth usage, and latency for multiplayer games. This will help players identify connection-related issues affecting gameplay.

Features:

- Measure ping and network bandwidth.
- Display real-time network traffic data.
- Alerts for high latency and packet loss.
- Explore cloud integration for remote latency tracking.

Phase 3: Game Optimization Module (Future Plans)

Summary: This phase will focus on detecting installed games using the Steam API and providing optimization recommendations based on the system's hardware and performance capabilities. The project may integrate NVIDIA API for advanced optimizations, such as AI-driven settings recommendations.

Features:

- Detect installed games via Steam API.
- Recommend optimal settings based on system specifications.
- Potential AI-driven recommendations using NVIDIA API (for RTX users).
- Implement cloud-based performance tuning, enabling users to store and retrieve recommended settings across devices.

Phase 4: Historical Data Analysis and Trends (Future Plans)

Summary: The final phase will focus on long-term data collection and analysis, allowing users to track system performance over time. This will be useful for identifying performance trends, pinpointing issues, and making better-informed optimizations.

Features:

- Store and visualize performance trends (e.g., FPS fluctuations over time).
- Allow users to export performance logs for in-depth analysis.
- Implement cloud storage support for syncing performance history across devices.

Risk Assessment

Identified Risks & Mitigation Strategies:

- Hardware Compatibility: Not all systems support certain monitoring tools.
    - Solution: Provide alternative metrics and fallback solutions.
- Third-Party Dependencies: Open Hardware Monitor may not work on all setups.
    - Solution: Develop custom PowerShell-based monitoring scripts.
- Scalability Issues: Real-time monitoring can use high resources.
    - Solution: Optimize backend performance and explore cloud-based processing.

Conclusion

The initial version of the Game Performance Monitoring Dashboard has successfully integrated the essential features such as:

- Monitor system performance in real time.
- Threshold breach alert system
- React-based UI for smooth visualization

Although some challenges (like fetching CPU temperature) needed some workarounds, the current system works fine. Going forward, Phase 2 I will add network latency tracking, and that will be followed by additional phases for monitoring multiple systems, performance optimization features, and cloud-based monitoring.

Through the ongoing developments, this project aims to turn into an all-inclusive, gamer-friendly performance optimizer that removes the necessity for third-party apps and boosts the gaming experience.