

**Project Proposal: Mod Master & GameTime Tracker**

### **1. Introduction**

Gaming plays an important role in the lives of today’s students, but juggling game modifications while keeping up with academic obligations can be difficult. This initiative presents Mod Master and GameTime Tracker, two desktop applications aimed at simplifying mod management and monitoring gaming duration, respectively. Both applications will utilize a common API, guaranteeing a smooth user experience and effective data management.

### **2. Problem Statement**

Many gamers struggle with:

* **Mod Management:** Installing, updating, and troubleshooting game mods can be complex and frustrating.
* **Time Management:** Uncontrolled gaming can interfere with academic performance and productivity.

### **3. Project Objectives**

This project aims to develop two interconnected applications:

#### **Mod Master:**

* Scan and detect installed game mods.
* Enable/disable mods with version tracking.
* Integrate with Nexus Mods for direct downloads.
* Log mod installations and detect conflicts.

#### **GameTime Tracker:**

* Monitor active game sessions and track playtime.
* Generate detailed reports and visual statistics.
* Allow users to set gaming limits and receive alerts.
* Integrate with study schedules for time management.

Both applications will share a common API for efficient data processing, ensuring:

* Centralized data handling for mod tracking and playtime management.
* Cross-application insights (e.g., tracking how modded games impact gaming hours).
* Scalable design for future enhancements.

### **4. Scope of Work**

The project will focus on the following key functionalities:

* **Mod Management System:** Scanning, enabling/disabling, and troubleshooting mods.
* **Gaming Time Tracking:** Detecting game processes and monitoring playtime.
* **Data Logging & Analysis:** Storing user activity for insights and troubleshooting.
* **API Development:** Creating a robust API for data exchange between both applications.
* **User Interface:** Designing intuitive interfaces using WPF or WinForms.

### **5. Expected Outcomes**

By the end of the project, we expect to deliver:

* A comprehensive mod management solution (*Mod Master*).
* A functional gaming time tracker (*GameTime Tracker*).
* A shared API facilitating communication between both applications.
* A user-friendly experience that enhances gaming while promoting balance.

### **6. Tools and Technologies**

* **Programming Language:** C#
* **Framework:** WPF or WinForms for GUI development
* **API Development:** REST API using ASP.NET Core
* **Database:** SQLite or JSON-based local storage
* **Logging & Monitoring:** Serilog for tracking user activity
* **Version Control:** GitHub for project management

### **7. APIs, Libraries, and Frameworks**

#### **Public API**

* **Nexus Mods API** – Used for fetching mod data, downloads, and version tracking.

*Mod Master will utilize the Nexus Mods API to:*

* Fetch mod details (name, version, author, etc.).
* Download and install mods directly.
* Check for mod updates.
* Ensure compatibility with installed mods.

*GameTime Tracker will use the Nexus Mod API To:*

* Correlate mod usage with gaming time.
* Identify which mods are used most frequently and their effect on session durations.
* Provide insights on whether certain mods contribute to increased playtime.

### **8. Conclusion**

The Mod Master & GameTime Tracker initiative introduces a unique method for managing mods and regulating gaming hours. By unifying these features through a common API, the project aims to develop an integrated and expandable system that improves the gaming experience while encouraging effective time management. The planning phase will lay out a clear framework, ensuring a well-defined scope and a productive development workflow.