



## **OS PROJECT PROPOSAL**

**Group Members:**

Sadia Tehreem (F23CSC043)

Syed Noor Ali Shah (F23CSC047)

**Course Instructor:**

*Miss Sumra Khan*

**Project Title:**

**“Multithreaded System Health Monitoring Tool”**

1. Introduction
2. Problem Statement
3. Objectives
4. Expected Outcomes
5. Conclusion

## **Introduction:**

This project is about creating a System Health Monitoring Tool that runs using multithreading. The main purpose is to check and display important system information like CPU usage, memory usage, disk space, and running processes in real time. By using multithreading, each task (like monitoring CPU or memory) runs in a separate thread, which makes the program faster and more efficient.

## **Problem Statement:**

Many system monitoring tools are heavy, slow, or not open-source. They often freeze when handling multiple tasks at once because everything runs in a single thread. There is a need for a lightweight and efficient tool that can monitor system performance smoothly by using multiple threads to handle different system resources at the same time.

## **Objectives:**

- >To design a tool that continuously monitors system health parameters like CPU, RAM, and disk usage.
- >To implement multithreading so that each monitoring function works independently.
- >To display real-time system status in a simple and user-friendly interface.
- >To understand and demonstrate how threads improve system performance and responsiveness.

## **Methodology:**

The project will be developed using Python (or C) programming language and multithreading techniques. Each system parameter (CPU, memory, disk, processes) will be monitored using a separate thread, so that all data updates happen in parallel. The program will use built-in libraries like psutil (Python) to fetch system data and display it using a text-based or GUI interface.

## **Expected Outcomes:**

- >A working multithreaded program that can monitor and display system performance in real time.
- >Faster and smoother monitoring compared to single-threaded tools.
- >Better understanding of how multithreading helps in efficient CPU utilization.
- >A simple interface where users can easily see system health details.

## **Modules:**

### **Module 1: CPU Monitoring**

- Displays CPU usage percentage and load per core.

### **Module 2: Memory Monitoring**

- Shows total, used, and available memory.

### **Module 3: Disk Usage Monitoring**

- Tracks disk space and read/write statistics.

### **Module 4: Process Monitoring**

- Lists active processes and their resource usage.

### **Module 5: Display Module**

- Combines all collected data and updates it live on screen.

## **Tools and Technologies:**

**Programming Language:** Python (or C)

**Operating System:** Windows / Linux

### **Libraries:**

- threading or multiprocessing (for threads)
- psutil (for system info)
- time, os, sys (for process handling)

**IDE:** VS Code or PyCharm

## **Work Plan:**

1 Research and planning, study multithreading concept

2 Set up development environment and collect tools

3 Implement CPU and memory monitoring threads

4 Add disk and process monitoring

5 Design display interface

6 Test and debug

7 Documentation and final report

**Conclusion:**

The **Multithreaded System Health Monitoring Tool** will show how using multiple threads can make an operating system task faster and more efficient.

It provides a practical example of real-time resource monitoring while learning about process management and concurrency.

This project combines both learning and usefulness — it's simple, effective, and demonstrates an important OS concept clearly.