📘 System Monitoring Tool – Documentation

🧠 Overview

This project is a lightweight and extensible system monitoring tool built using Bash scripts. It collects key system metrics (CPU, memory, disk, GPU, etc.), supports both CLI and HTML reporting, and runs inside a Docker container for portability and consistency across environments.

📁 Project Structure

```
system-monitor/
│
├── monitor.sh            # Main system monitoring script
├── html_report.sh        # Generates styled HTML reports
├── gui_monitor.sh        # CLI-based menu for easy interaction
├── dashboard.sh          # Dashboard To View TxT Logs
├── Dockerfile            # Builds the Docker image
├── docker-compose.yml    # Runs the container
├── logs/                 # Contains generated .txt reports
└── html_logs/            # Contains generated HTML reports
```

⚙️ How It Works

🔹 1. `monitor.sh`

This is the main script that performs system monitoring. It collects:

- **CPU Performance** using `mpstat`
- **CPU Temperature** using `sensors` (if available)
- **GPU Info** using `nvidia-smi` (if NVIDIA GPU is present)
- **Memory Usage** using `free -h`
- **Disk Usage** using `df -h`
- **SMART Disk Status** using `smartctl` (optional)
- **Network Interfaces & IPs** using `ip a`
- **System Load** using `uptime`
- **User and Host Information** using `whoami` and `hostname`

💾 It automatically creates a report file in the `logs/` directory with a timestamped filename.

🔹 **2.** `html_report.sh`

- Generates an **HTML version** of the report, with colors and basic formatting.
- Saves the report in `html_logs/` with a timestamped filename.
- Useful for viewing system reports in a web browser.

🔹 **3.** `gui_monitor.sh`

- A simple **menu-based interface** using basic Bash prompts.
- Allows the user to:

```
1) Generate Full System Report
2) Generate HTML Report
3) Exit
```

- Helps users who don't want to type commands manually.

🐳 Docker Integration

🔸 `Dockerfile`

- Based on **Ubuntu 24.04**.
- Installs necessary packages: `dialog`, `util-linux`, `net-tools`, `procps`, `curl`, `nano`, etc.
- Copies your scripts into the container and makes them executable.

🔸 `docker-compose.yml`

- Simplifies running the container using a single command:

```
Bash:
docker-compose up --build
```

- Creates the required network and environment automatically.

🧩 This makes your monitoring tool **portable** — it can run the same way on any system with Docker installed, no manual setup needed.

🛠️ Setup Instructions

🧪 Run Locally (WSL or Linux):

```
Bash:
chmod +x monitor.sh
./monitor.sh
```

🖥️ GUI Version:

```
Bash:
chmod +x gui_monitor.sh
./gui_monitor.sh
```

🌐 HTML Report:

```
Bash:
chmod +x html_report.sh
./html_report.sh
```

🖥️ Dashboard Version:

```
Bash:
chmod +x dashboard.sh
./dashboard.sh
```

🐳 Run in Docker:

```
Bash:
docker-compose up --build
```

📄 Output Examples

- **Text report:** `logs/report_YYYY-MM-DD_HH-MM-SS.txt`
- **HTML report:** `html_logs/report_YYYY-MM-DD_HH-MM-SS.html`

🔍 Additional Notes

- If `sensors` doesn't work, you may not have temperature sensors or you're using WSL.
- `nvidia-smi` requires NVIDIA GPU and driver support.
- The tool gracefully skips unsupported features.