🖥️ Advanced System Health Monitor

Enterprise-Grade Real-Time System Diagnostics Dashboard

A powerful, feature-rich Python application for comprehensive system monitoring with real-time metrics, visual analytics, and export capabilities.

✨ Features Overview

📊 Real-Time Monitoring

- **Live CPU Tracking** - Per-core analysis with frequency and temperature monitoring

- **Memory Management** - RAM and SWAP memory usage statistics

- **Disk Analysis** - Storage capacity and I/O performance metrics

- **Network Diagnostics** - Real-time bandwidth monitoring

- **Battery Status** - Battery percentage, charging state, and time remaining

- **System Information** - OS details, uptime, and hardware specifications

🎨 Interactive Dashboard

- **Multi-Tab Interface** - Organized data across 7 dedicated tabs

- **Visual Indicators** - Color-coded health status (Green/Blue/Orange/Red)

- **Dynamic Overview** - At-a-glance dashboard with key metrics

- **Dark Theme** - Eye-friendly interface with professional styling

💾 Export Capabilities

- **JPEG Export** - High-quality image snapshots of reports

- **PNG Export** - Lossless image format for archiving

- **Timestamped Reports** - Automatic naming with date/time

🔧 Advanced Monitoring

- **Customizable Thresholds** - Define health status boundaries

- **Continuous Updates** - Auto-refresh every 15 seconds

- **Performance Metrics** - Read/write speeds and network throughput

- **Thermal Monitoring** - CPU temperature tracking (where available)

🚀 Quick Start

Prerequisites

```bash
# Python 3.7+
python --version

# Required packages:
pip install psutil pillow
```

Installation

```bash
# 1. Clone or download the project
git clone <repository-url>
cd system-health-monitor

# 2. Install dependencies
pip install -r requirements.txt

# 3. Run the application
python system_health_monitor.py
```
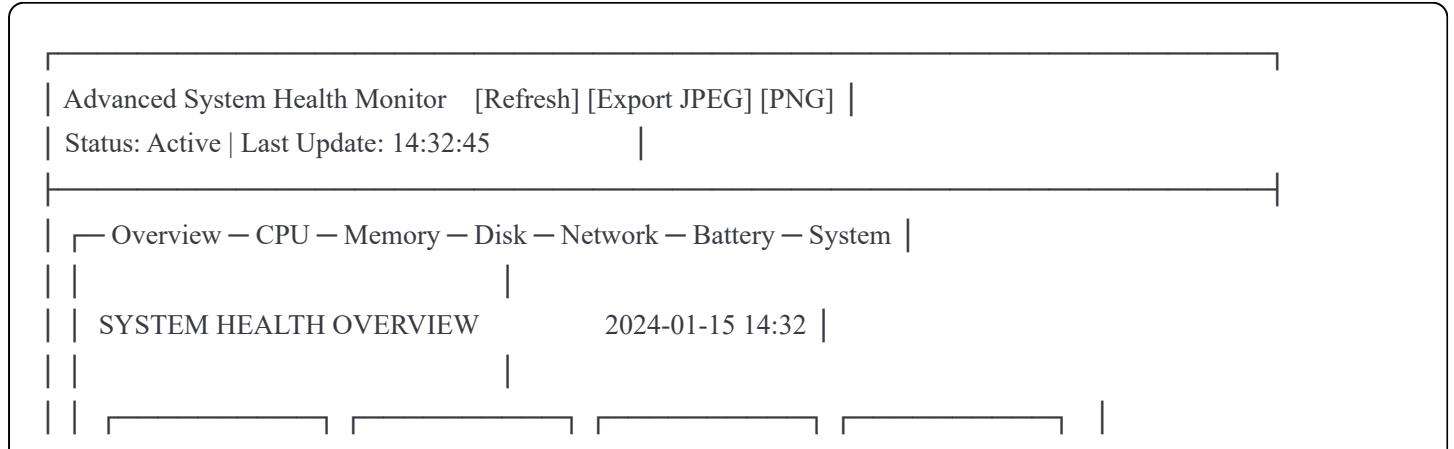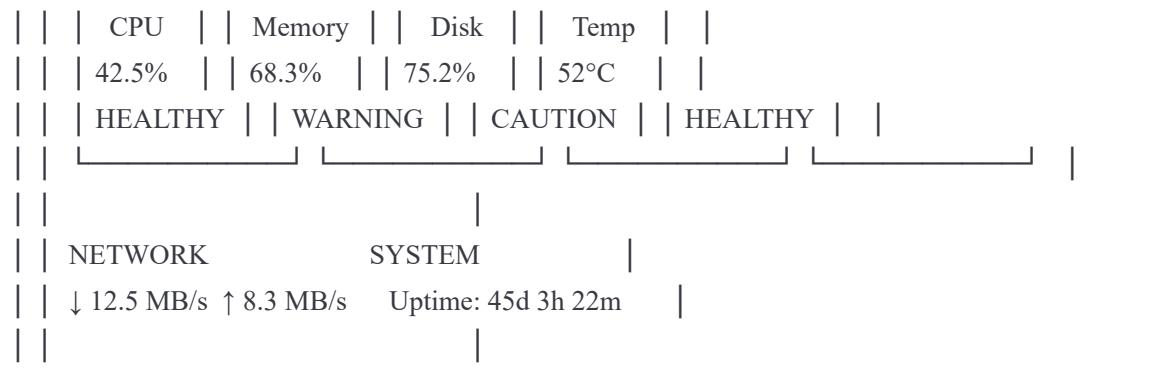
Requirements File

Create `requirements.txt`:

```
psutil>=5.9.0
Pillow>=8.0.0
```

📋 Interface Guide

Main Window Layout

```
┌─────────────────────────────────────────────────┐
│ Advanced System Health Monitor   [Refresh] [Export JPEG] [PNG] │
│ Status: Active | Last Update: 14:32:45        │              │
├─────────────────────────────────────────────────┤
│ ┌─ Overview ─ CPU ─ Memory ─ Disk ─ Network ─ Battery ─ System │
│ │                              │
│ │  SYSTEM HEALTH OVERVIEW         2024-01-15 14:32 │
│ │                              │
│ │ ┌────────┐  ┌────────┐  ┌────────┐  ┌────────┐ │
```

```
| |  | CPU   | | Memory | | Disk   | | Temp  | | |
| |  | 42.5% | | 68.3%  | | 75.2%  | | 52°C  | | |
| |  | HEALTHY | | WARNING | | CAUTION | | HEALTHY | | |
| |  └────────┘ └─────────┘ └─────────┘ └────────┘   |
| |                         |
| |  NETWORK        SYSTEM              |
| |  ↓ 12.5 MB/s  ↑ 8.3 MB/s    Uptime: 45d 3h 22m     |
| |                         |
| |  └──────────────────────────────────────────┘
```

Tab Details

🔴 Overview Tab

- Visual metric boxes with color-coded status

- Real-time values for CPU, Memory, Disk, and Temperature

- Network speed indicators

- System uptime information

🔧 CPU Tab

```
Status:        HEALTHY
Average Usage:   42.50%
CPU Cores:       8
Logical Threads: 16
Frequency:       2400.00 MHz
Temperature:     52.30°C


Per-Core Usage:
Core 0: ███████░░░░░░░░░░░░░░░  35.20%
Core 1: █████░░░░░░░░░░░░░░░░░  28.50%
...
```

💾 Memory Tab

```
Status:        WARNING
RAM Usage:      68.30%


RAM Details:
Total:      16.00 GB
Used:       10.93 GB (68.3%)
Available:   5.07 GB


Visual:      ████████░░░░░
```

📁 Disk Tab

Status: CAUTION

Disk Usage: 75.20%

Capacity Details:

Total: 500.00 GB

Used: 375.95 GB (75.2%)

Free: 124.05 GB

I/O Performance:

Read Speed: 245.50 MB/s

Write Speed: 156.80 MB/s

🌐 Network Tab

Real-Time Speeds:

Download: 12.50 MB/s

Upload: 8.30 MB/s

Network Interfaces:

eth0   Status: UP   MTU: 1500

wlan0   Status: DOWN MTU: 1500

🔋 Battery Tab

Status: HEALTHY

Charge Level: 85.5%

Charging Status: Charging

Time Remaining: 3:45:00

Visual:

ℹ️ System Tab

System Details:

Hostname:　　　WORKSTATION-01

Operating System: Windows 11 22621

Architecture:　　64bit

Processor:　　Intel Core i7-10700K

Uptime Information:

Last Boot:　　2024-01-01 08:15:30

Current Uptime:　45d 3h 22m

🎯 Health Status Levels

Color Coding System

Status	Color	Threshold	Meaning
🟢 **HEALTHY**	Green	Safe range	System operating normally
🔵 **CAUTION**	Blue	Mild concern	Monitor for changes
🟠 **WARNING**	Orange	Moderate concern	Action recommended
🔴 **CRITICAL**	Red	Severe concern	Immediate attention needed
🟣 **UNKNOWN**	Gray	N/A	Data unavailable

Default Thresholds

CPU Usage:
 • Healthy:　0-60%
 • Caution:　60-80%
 • Warning:　80-90%
 • Critical:　90%+

Memory/Disk:
 • Healthy:　0-70%
 • Caution:　70-85%
 • Warning:　85-95%
 • Critical:　95%+

Battery:
 • Healthy:　>50%
 • Caution:　20-50%
 • Warning:　10-20%
 • Critical:　<10%

Temperature:
 • Healthy:　<70°C

- Caution: 70-85°C
- Warning: 85-100°C
- Critical: >100°C

🎮 Usage Guide

Basic Operations

Force Refresh

Click the `Refresh` button to immediately update all metrics without waiting for the 15-second cycle.

Export Dashboard

1. Click `Export as JPEG` or `Export as PNG`

2. Choose save location

3. Report generates automatically with timestamp

Monitor in Real-Time

- Application auto-updates every 15 seconds

- Status indicator shows last update time

- Click tabs to view detailed information

Export Report Format

Generated reports include:

- 📅 Timestamp of generation

- 📊 All system metrics at export time

- 🎯 Health status indicators

- 💻 Hardware specifications

- 📈 Performance data

Filename Pattern: `system_health_YYYYMMDD_HHMMSS.jpg/png`

⚙️ Configuration

Customizing Thresholds

Edit the `THRESHOLDS` dictionary in the code:

```
python
```

```
THRESHOLDS = {
    'cpu_critical': 90,        # Adjust CPU critical level
    'memory_warning': 85,      # Adjust memory warning level
    'disk_caution': 70,        # Adjust disk caution level
    'temp_critical': 100,      # Adjust temperature critical
    'battery_critical': 10,    # Adjust battery critical
}
```

Update Interval

Change monitoring refresh rate (default: 15 seconds):

```python
python

time.sleep(15)  # Modify this value in monitor_loop()
```

📊 System Requirements

Minimum Requirements

- **Python:** 3.7 or higher

- **RAM:** 512 MB

- **Disk Space:** 50 MB

- **OS:** Windows, macOS, Linux

Recommended

- **Python:** 3.9+

- **RAM:** 2 GB

- **Processor:** Multi-core CPU

Supported Operating Systems

- ✅ Windows 7/10/11

- ✅ macOS 10.14+

- ✅ Linux (Ubuntu, Fedora, Debian, etc.)

🔧 Advanced Features

Temperature Monitoring

- Automatically detects CPU temperature sensors

- Supports Intel and AMD processors

- Falls back gracefully if unavailable

Network Analysis

- Real-time bandwidth calculation

- Interface status tracking

- Bytes sent/received monitoring

Disk I/O Metrics

- Read/write speed calculation

- Byte counter accumulation

- Performance trending

Battery Analytics

- Charge percentage tracking

- AC adapter detection

- Time-to-empty estimation

🐛 Troubleshooting

Common Issues

Issue: Application won't start

```
Solution: Ensure all dependencies installed
pip install --upgrade psutil pillow
```

Issue: Temperature shows as N/A

```
Solution: Normal on some systems; requires lm-sensors (Linux)
sudo apt-get install lm-sensors  # Linux
```

Issue: Export fails

```
Solution: Check folder permissions and disk space
Ensure write access to save directory
```

Issue: High CPU usage

Solution: Increase monitoring interval or close other apps
Edit time.sleep(15) to time.sleep(30) for less frequent updates

---

📁 File Structure

```
system-health-monitor/
├── system_health_monitor.py   # Main application
├── requirements.txt          # Dependencies
├── README.md                # Documentation
└── exports/                # Export folder (auto-created)
   ├── system_health_*.jpg
   └── system_health_*.png
```

🎨 UI Customization

Theme Colors

Modify the `HealthStatus` class:

```python
COLOR_CRITICAL = "#FF4444"    # Red
COLOR_WARNING = "#FFAA00"     # Orange
COLOR_HEALTHY = "#44AA44"     # Green
COLOR_BG = "#1E1E1E"         # Background
COLOR_TEXT = "#E0E0E0"       # Text
```

Font Sizes

Adjust in `create_*_tab()` methods:

```python
font=("Courier", 10)  # Change size as needed
```

📈 Performance Tips

1. **Reduce Update Frequency** - Increase interval for lower CPU usage

2. **Close Unused Tabs** - Focus on needed information only

3. **Disable Temperature** - Skip sensor reading if not needed

4. **Batch Operations** - Export multiple reports in one session

5. **Regular Cleanup** - Delete old export files periodically

---

🔐 Security & Privacy

- **Local Processing** - No data sent to external servers

- **No Logging** - Metrics not stored permanently

- **User Controlled** - You decide what to export and where

- **No Network Calls** - Operates completely offline

📝 Keyboard Shortcuts

Shortcut	Action
Tab	Switch between tabs
Ctrl+R	Manual refresh (Refresh button)
Ctrl+E	Export menu
Ctrl+Q	Quit application
F5	Refresh current data

🤝 Contributing

Found a bug? Have suggestions? Please feel free to:

1. Report issues with detailed system information

2. Suggest new features or metrics

3. Contribute code improvements

📜 License

This project is provided as-is for personal and professional use.

📞 Support

For issues:

- Check the Troubleshooting section

- Verify all dependencies installed

- Ensure compatible Python version

- Check system permissions

For feature requests:

- Document your use case

- Explain the desired functionality

- Provide examples if possible

🎯 Roadmap

Planned Features:

- 📊 Historical data graphing

- 🔔 Alert notifications

- 📤 Cloud backup integration

- 🔒 System security scanning

- 📱 Mobile companion app

💡 Pro Tips

✅ Best Practices:

- Export reports during peak usage for accurate baseline

- Monitor temperatures during intensive workloads

- Use JPEG for emails, PNG for archiving

- Check battery health monthly

- Review disk usage trends weekly

⚠️ When to Take Action:

- CPU consistently >80% → Close unnecessary apps

- Memory >90% → Upgrade or clear cache

- Disk >85% → Delete files or expand storage

- Temperature >85°C → Clean fans or improve cooling

- Battery degradation → Consider replacement

🙏 Acknowledgments

Built with:

- **psutil** - System and process utilities

- **Pillow** - Image processing

- **tkinter** - GUI framework

⭐ If you find this useful, consider starring the repository!

Advanced System Health Monitor - Your Complete System Diagnostics Solution