SSPL Central Network Monitoring System

Step-by-Step Setup and User Guide

For Internal Use at SSPL/DRDO

June 24, 2025

Introduction

The SSPL Central Network Monitoring System is a powerful tool designed to collect and visualize hardware and software inventory from multiple Windows and Linux machines. It features a central dashboard, downloadable reports, and simple agents for both operating systems. This guide will walk you through the setup and usage of the system in a detailed, step-by-step manner.

Note: This system can be used entirely offline within your organization's intranet. No internet connection is required after initial setup.

1 System Features

- Collects detailed hardware and software information from Windows and Linux clients
- Centralized dashboard for device inventory
- Download device data as CSV or PDF
- Simple login-protected web interface
- Agents for Windows (PowerShell) and Linux (Bash)
- Can be used offline in a secure intranet environment

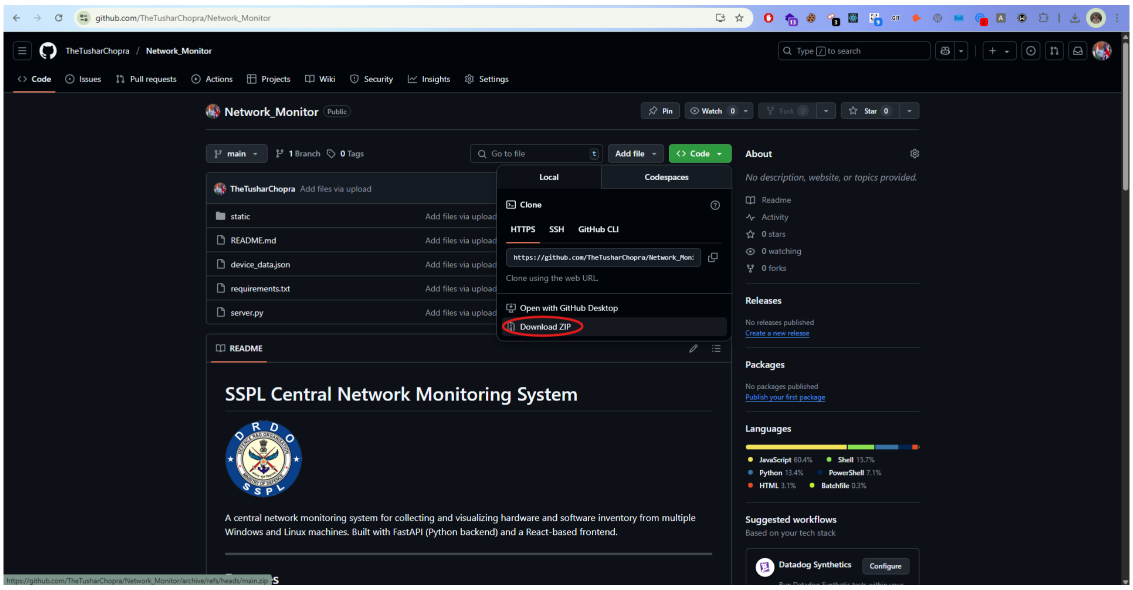
2 Prerequisites

- **Python 3.8+** (for the server)
- **pip** (Python package manager)
- **For Linux agent:** curl, lsb_release, dmidecode, lsblk, awk, etc.
- **For Windows agent:** PowerShell 5+

3 Step 1: Downloading the Project from GitHub

This project is hosted at:
`https://github.com/TheTusharChopra/Network_Monitor`

For Windows Users

1. Open your web browser and go to: `https://github.com/TheTusharChopra/Network_Monitor`
2. Click the green **Code** button and select **Download ZIP**.
3. Save the ZIP file to your computer and extract it to a folder, e.g., `C:Network_Monitor`.
4. 

SSPL Central Network Monitoring System

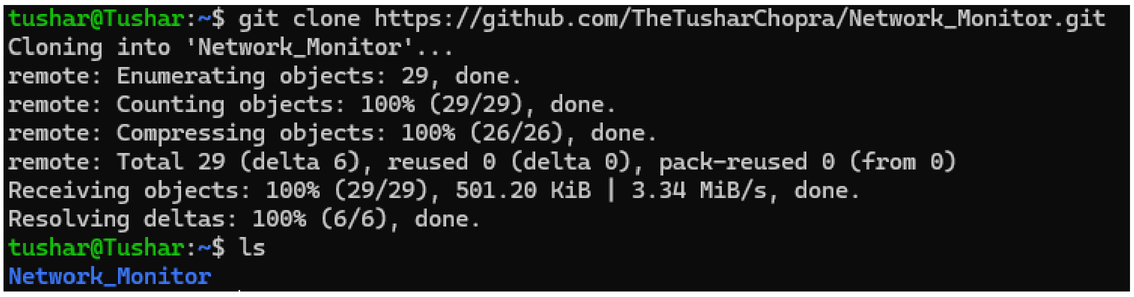
A central network monitoring system for collecting and visualizing hardware and software inventory from multiple Windows and Linux machines. Built with FastAPI (Python backend) and a React-based frontend.

For Linux Users

1. Open a terminal window.
2. Run the following command to download the project:

```
git clone https://github.com/TheTusharChopra/Network_Monitor.git
```

3. The project will be downloaded to a folder named `Network_Monitor` in your current directory.

4. 

```
tushar@Tushar:~$ git clone https://github.com/TheTusharChopra/Network_Monitor.git
Cloning into 'Network_Monitor'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 29 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (29/29), 501.20 KiB | 3.34 MiB/s, done.
Resolving deltas: 100% (6/6), done.
tushar@Tushar:~$ ls
Network_Monitor
```

4 Step 2: Setting Up Python Environment

For Windows

1. Open the extracted project folder (e.g., `C:`
`Network_Monitor`).

2. Right-click inside the folder and select **Open in Terminal** or **Open PowerShell window here**.

3. (Recommended) Create a virtual environment:

```
python -m venv venv
```

4. Activate the virtual environment:

```
venv\Scripts\activate
```

5. Install the required Python packages:

```
pip install -r requirements.txt
```

For Linux

1. Open a terminal and navigate to the project directory:

```
cd Network_Monitor
```

2. (Recommended) Create a virtual environment:

```
python3 -m venv venv
```

3. Activate the virtual environment:

```
source venv/bin/activate
```

4. Install the required Python packages:

```
pip install -r requirements.txt
```

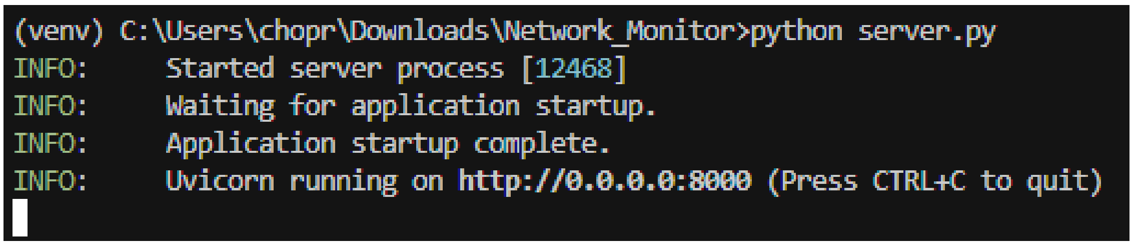
5 Step 3: Running the Server

For Windows

1. In the terminal (with the virtual environment activated), run:

```
python server.py
```

2. The server will start and listen on `http://0.0.0.0:8000/` (default port 8000).

3. 

```
(venv) C:\Users\chopr\Downloads\Network_Monitor>python server.py
INFO:     Started server process [12468]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

4. Leave this terminal window open while the system is in use.

For Linux

1. In the terminal (with the virtual environment activated), run:

```
python3 server.py
```

2. The server will start and listen on `http://0.0.0.0:8000/` (default port 8000).

3. Leave this terminal window open while the system is in use.

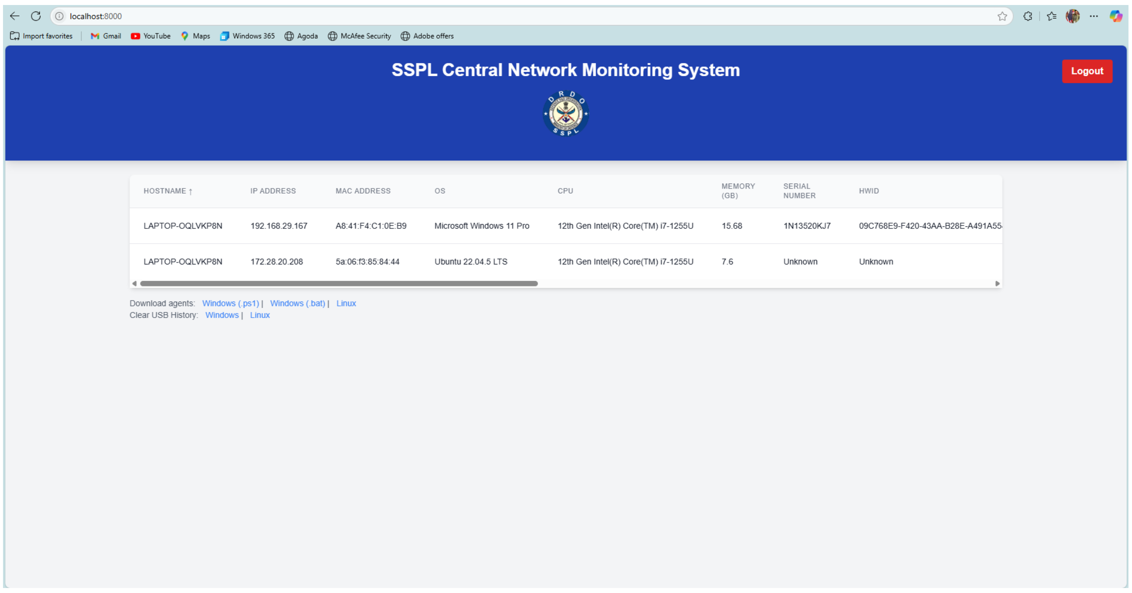
6 Step 4: Accessing the Web Dashboard

1. Open a web browser on any computer connected to the same network (intranet).

2. IP of the server computer can be found by typing 'ipconfig' in terminal for windows and 'ifconfig' in linux terminal for linux.

3. Enter the server's IP address and port in the address bar, for example:

```
http://<server-ip>:8000/
```

4. You will see the login page. Use the following credentials:
   - **Username:** sspl
   - **Password:** password
5. These credentials can be changed in app.js as it has been stored locally.
6. After logging in, you will see the dashboard with all reported devices.
7. 

   localhost:8000

   SSPL Central Network Monitoring System

HOSTNAME ↑	IP ADDRESS	MAC ADDRESS	OS	CPU	MEMORY (GB)	SERIAL NUMBER	HWID
LAPTOP-OQLVKP8N	192.168.29.167	A8:41:F4:C1:0E:B9	Microsoft Windows 11 Pro	12th Gen Intel(R) Core(TM) i7-1255U	15.68	1N13520KJ7	09C768E9-F420-43AA-B28E-A491A55
LAPTOP-OQLVKP8N	172.28.20.208	5a:06:f3:85:84:44	Ubuntu 22.04.5 LTS	12th Gen Intel(R) Core(TM) i7-1255U	7.6	Unknown	Unknown

   Download agents: Windows (.ps1) | Windows (.bat) | Linux
   Clear USB History: Windows | Linux

8. OPEN THE AGENTS (BOTH POWERSHELL AND SH FILES) AND REPLACE ALL EXISTING IP WITH THE CURRENT SERVER IP EVERYWHERE.

7 Step 5: Deploying Agents on Client Machines

Windows Agent

1. Download `agent.ps1` and `runagent.bat` from `https://<server-ip>:8000/download/windows` and `https://<server-ip>:8000/download/windows-bat`
2. Make sure both files are in the same folder
3. To run the agent:
   - Double click runagent.bat
   - If you see "Windows SmartScreen Protected your PC", Click on More Info and then Run Anyway.
   - Wait until you see Data and Logs sent successfully on the command prompt.
   - Once the data is sent, press Ctrl+C it will ask "Terminate Batch Job?" Type "Y" and press enter.
4. The agent will collect system information and send it to the server. If the server is unreachable, it will retry automatically.

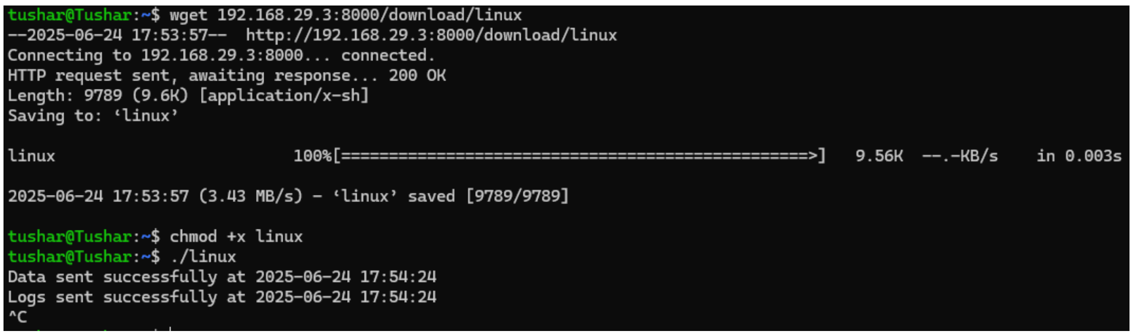
Linux Agent

1. Download `agent.sh` using `wget <server-ip>:8000/download/linux`
2. Make the script executable:

   ```
   chmod +x <filename>
   ```

3. Run the script:

   ```
   ./<filename>
   ```

4. The agent will collect system information and send it to the server. If the server is unreachable, it will retry automatically.
5. 

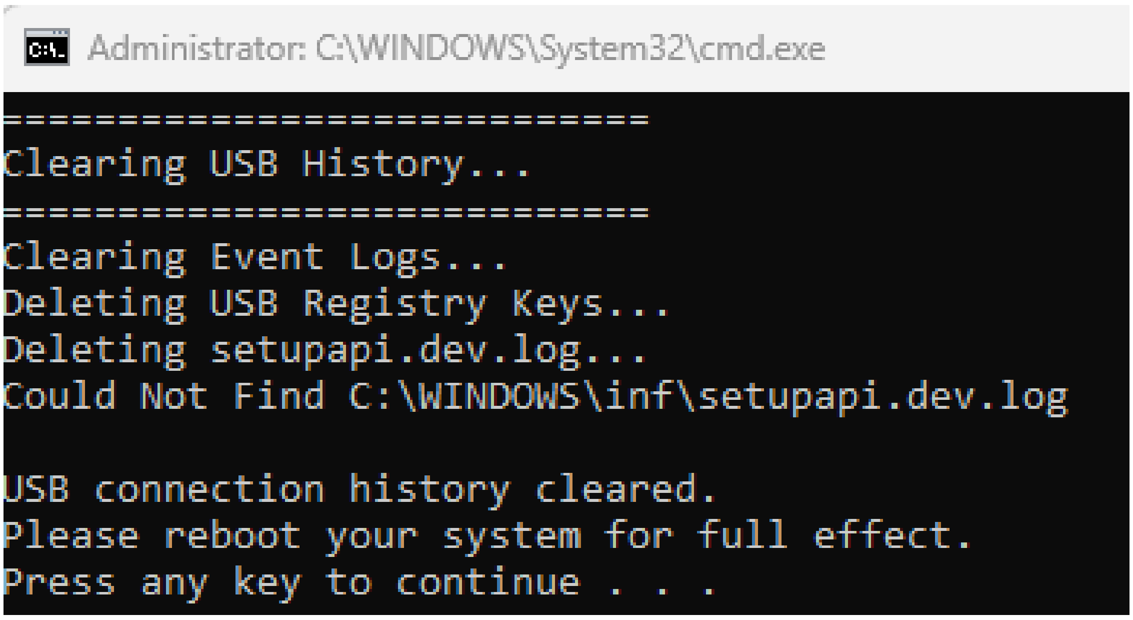
```
tushar@Tushar:~$ wget 192.168.29.3:8000/download/linux
--2025-06-24 17:53:57--  http://192.168.29.3:8000/download/linux
Connecting to 192.168.29.3:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9789 (9.6K) [application/x-sh]
Saving to: ‘linux’

linux                         100%[=================================================>]   9.56K  --.-KB/s    in 0.003s

2025-06-24 17:53:57 (3.43 MB/s) - ‘linux’ saved [9789/9789]

tushar@Tushar:~$ chmod +x linux
tushar@Tushar:~$ ./linux
Data sent successfully at 2025-06-24 17:54:24
Logs sent successfully at 2025-06-24 17:54:24
^C
```

6. Once logs and data is sent successfully, press Ctrl+C to close the process.

8 Step 6: Clearing USB History

Windows Agent

1. Download `clear_usb_history.bat` from `https://<server-ip>:8000/download/clear_usb_history_windows`
2. Double click the downloaded file.
3. It would request admin privileges, click yes.
4. Once the USB history is cleared, click any key to close the terminal.

5. 

```
Administrator: C:\WINDOWS\System32\cmd.exe
================================
Clearing USB History...
================================
Clearing Event Logs...
Deleting USB Registry Keys...
Deleting setupapi.dev.log...
Could Not Find C:\WINDOWS\inf\setupapi.dev.log

USB connection history cleared.
Please reboot your system for full effect.
Press any key to continue . . .
```

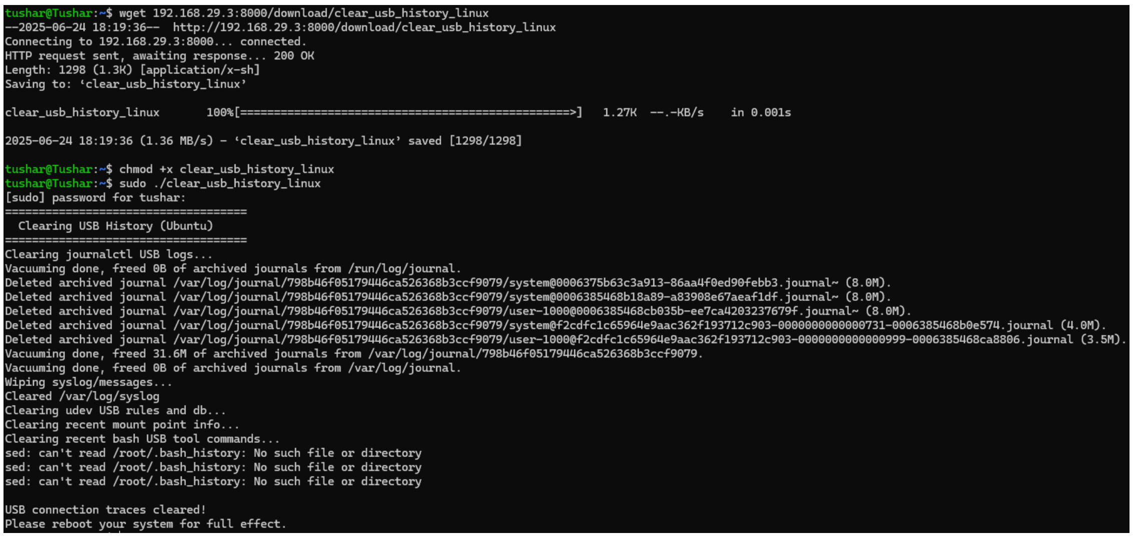
Linux Agent

1. Download `clear_usb_history.sh` from `https://<server-ip>:8000/download/clear_usb_history_linux`
2. Make the script executable:

```
chmod +x clear_usb_history_linux
```

3. Run the script:

```
sudo ./clear_usb_history_linux
```

4. This agent will clear all the USB logs history from the system.
5. 

```
tushar@Tushar:~$ wget 192.168.29.3:8000/download/clear_usb_history_linux
--2025-06-24 18:19:36--  http://192.168.29.3:8000/download/clear_usb_history_linux
Connecting to 192.168.29.3:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1298 (1.3K) [application/x-sh]
Saving to: ‘clear_usb_history_linux’

clear_usb_history_linux       100%[=============================================>]   1.27K  --.-KB/s    in 0.001s

2025-06-24 18:19:36 (1.36 MB/s) - ‘clear_usb_history_linux’ saved [1298/1298]

tushar@Tushar:~$ chmod +x clear_usb_history_linux
tushar@Tushar:~$ sudo ./clear_usb_history_linux
[sudo] password for tushar:
====================================
  Clearing USB History (Ubuntu)
====================================
Clearing journalctl USB logs...
Vacuuming done, freed 0B of archived journals from /run/log/journal.
Deleted archived journal /var/log/journal/798b46f05179446ca526368b3ccf9079/system@0006375b63c3a913-86aa4f0ed90febb3.journal~ (8.0M).
Deleted archived journal /var/log/journal/798b46f05179446ca526368b3ccf9079/system@0006385468b18a89-a83908e67aeaf1df.journal~ (8.0M).
Deleted archived journal /var/log/journal/798b46f05179446ca526368b3ccf9079/user-1000@0006385468cb035b-ee7ca4203237679f.journal~ (8.0M).
Deleted archived journal /var/log/journal/798b46f05179446ca526368b3ccf9079/system@f2cdfc1c65964e9aac362f193712c903-0000000000000731-0006385468b0e574.journal (4.0M).
Deleted archived journal /var/log/journal/798b46f05179446ca526368b3ccf9079/user-1000@f2cdfc1c65964e9aac362f193712c903-0000000000000999-0006385468ca8806.journal (3.5M).
Vacuuming done, freed 31.6M of archived journals from /var/log/journal/798b46f05179446ca526368b3ccf9079.
Vacuuming done, freed 0B of archived journals from /var/log/journal.
Wiping syslog/messages...
Cleared /var/log/syslog
Clearing udev USB rules and db...
Clearing recent mount point info...
Clearing recent bash USB tool commands...
sed: can't read /root/.bash_history: No such file or directory
sed: can't read /root/.bash_history: No such file or directory
sed: can't read /root/.bash_history: No such file or directory

USB connection traces cleared!
Please reboot your system for full effect.
```

9 Admin Dashboard Functionalities

- **View Devices:** All reported devices are listed in a sortable table.
- **Download Data:** Download device data as CSV or PDF for offline analysis or record-keeping.
- **USB History:** See the last time a USB device was connected to each machine in the Downloaded PDF.
- **Download Agents:** Download Windows and Linux agent scripts directly from the dashboard.
- **Clear USB History:** Download scripts to clear USB history on client machines.

10 Accessible Routes of the Application

- `/` — Main dashboard (web UI)
- `/devices` — API endpoint to get all device data (JSON)
- `/devices/report` — API endpoint for agents to report device data (POST)
- `/devices/logs` — API endpoint for agents to report logs (POST)
- `/devices/logs/ip_address` — API endpoint to get logs for a device
- `/download/windows` — Download Windows agent script
- `/download/linux` — Download Linux agent script
- `/download/windows-bat` — Download Windows batch wrapper
- `/download/clear_usb_history_windows` — Download Windows USB clear script
- `/download/clear_usb_history_linux` — Download Linux USB clear script

11 Technology Stack

- **Backend:** Python 3.8+, FastAPI, Uvicorn
- **Frontend:** React (served as static files), HTML, CSS (Tailwind)
- **Agents:** PowerShell (Windows), Bash (Linux)
- **Data Storage:** JSON files (`device_data.json`, `device_logs.json`)

12 How Does This Run Offline?

- The entire system is self-contained and does not require internet access after initial download.
- All communication between server, dashboard, and agents happens within your local network (intranet).
- Data is stored locally on your server; nothing is sent outside your organization.
- You can deploy and use the system in secure, air-gapped environments.

13 Troubleshooting

- **Server not starting:** Ensure Python 3.8+ is installed and all dependencies are present. Check for errors in the terminal.
- **Cannot access dashboard:** Make sure the server is running and the firewall allows connections on port 8000.
- **Agent not reporting:** Ensure the agent script is run as administrator/root and the server IP is correctly set in the script.
- **Data not updating:** Refresh the dashboard page. Check the server terminal for errors.

14 Security and Privacy

- All data is stored locally on your server. No information is sent outside your network.
- Only authorized users with the correct credentials can access the dashboard.
- Agents cache data locally and only send it to your server.

Appendix: File Structure

- `server.py` — FastAPI backend
- `static/` — Frontend (React), agents, and assets
- `device_data.json` — Device data cache (auto-created)
- `requirements.txt` — Python dependencies

This project is for internal use at SSPL/DRDO.