**Project OS**

**Documentation**

**BY:**

1. Gasser Amr

System Information Reporting Project - Detailed Documentation

This documentation provides a comprehensive overview of the system reporting project. It covers the project structure, detailed functionality of each function and variable, installed packages, HTML and CSS design, and step-by-step instructions for running the project using Zenity and Docker.

Project Overview

This project generates system reports (CPU, disk, memory, network, GPU, and system load) in HTML format. The reports are styled using embedded CSS and displayed using Zenity dialogs. The project supports execution both on the host system and within a Docker container.

Directory Structure

* gasser/project.sh: The main script that generates reports.
* gasser/\*.html: HTML reports generated by the script.
* gasser/system\_report.log: Log file recording events.
* Dockerfile: Docker configuration file for containerizing the project.

Code Explanation

**Main Script (project.sh)**

**Variables**

* LOGFILE: Path to the log file where events are recorded.

**Functions**

1. **log\_event()**
   * Logs a message to the system\_report.log file with a timestamp.
   * **Parameters**:
     + message: The event message to log.
   * **Example**:
   * log\_event "CPU report generated"
2. **generate\_html\_report()**
   * Creates a styled HTML report using a template.
   * **Parameters**:
     + title: The title of the report.
     + content: The main content of the report (HTML format).
     + filename: The output filename for the report.
   * **HTML/CSS Details**:
     + **HTML**:
       - Includes sections like <head>, <body>, <h1> for title, <table> for data, and <footer> for timestamp.
     + Example table structure:
     + <table>
     + <tr><th>Header</th></tr>
     + <tr><td>Data</td></tr>
     + </table>
     + **CSS**:
       - Styled with colors, padding, and responsive design.
       - Example:
       - body {
       - font-family: Arial, sans-serif;
       - background-color: #f4f4f9;
       - }
3. **cpu\_info()**
   * Gathers CPU details using lscpu and performance stats via top.
   * **Output**: Generates cpu\_report.html and logs the event.
   * Example content:
     + CPU Model, Speed, and performance metrics.
4. **disk\_info()**
   * Retrieves disk usage with df -h and SMART health status using smartctl.
   * **Output**: Generates disk\_report.html with details like:
     + Filesystem size, usage, and health status.
5. **memory\_info()**
   * Displays memory statistics using free -h.
   * **Output**: Generates memory\_report.html.
6. **network\_info()**
   * Collects network interface details with ip a and statistics using netstat.
   * **Output**: Generates network\_report.html.
7. **gpu\_info()**
   * Detects GPU info using tools like nvidia-smi (NVIDIA) or amdgpu-smi (AMD).
   * **Output**: Generates gpu\_report.html.
8. **system\_load()**
   * Reports system load, CPU, memory, and swap usage via uptime and top.
   * **Output**: Generates system\_load\_report.html.
9. **show\_menu()**
   * Displays a Zenity-based menu for user interaction.
   * **Options**:
     + Generate reports for CPU, disk, memory, network, GPU, or system load.
     + Exit the script

**Main Loop**

* Repeatedly shows the menu until the user selects "Exit".

Dockerfile Explanation

1. **Base Image**
2. FROM ubuntu:latest
   * Uses the latest Ubuntu image as the base.
3. **Working Directory**
4. WORKDIR /app
   * Sets /app as the working directory inside the container.
5. **Dependencies Installation**
6. RUN apt-get update && apt-get install -y \
7. zenity \
8. smartmontools \
9. iproute2 \
10. net-tools \
11. procps \
12. && rm -rf /var/lib/apt/lists/\*
    * Installs required packages for the project.
13. **Copy Files**
14. COPY gasser /app/gasser
    * Copies the gasser directory to the container.
15. **Set Executable Permissions**
16. RUN chmod +x /app/gasser/project.sh
    * Makes the script executable.
17. **Default Command**
18. CMD ["/bin/bash", "/app/gasser/project.sh"]
    * Runs the script when the container starts.

Required Packages

* **Zenity**: For GUI dialogs.
* **Smartmontools**: For disk health monitoring.
* **Net-tools**: For network statistics.
* **Procps**: For process information.
* **IPRoute2**: For IP configuration and routing.

Usage Instructions

**Using Zenity Only**

1. Ensure dependencies are installed:
2. sudo apt-get update && sudo apt-get install -y zenity smartmontools iproute2 net-tools procps
3. Run the script:
4. ./gasser/project.sh

**Using Docker**

1. Build the Docker image:
2. docker build -t system\_report\_image .
3. Run the Docker container:
4. docker run -it system\_report\_image

**Commands Summary**

* **Build Image**:
* docker build -t system\_report\_image .
* **Run Container**:
* docker run -it system\_report\_image
* run -e DISPLAY=$DISPLAY --net=host -v /tmp/.X11-unix:/tmp/.X11-unix
* **Execute Script Locally**:
* ./gasser/project.sh

This documentation provides all necessary details for understanding, running, and extending the project. If further clarification is needed, feel free to ask!