Project Report: Automating System Management Tasks

**Introduction**

This project focuses on automating essential system management tasks using Bash scripting in a Linux environment. The tasks include log file management, system health monitoring, and software installation automation. The deliverables showcase the practical applications of Bash scripting to streamline system administration processes.

**Task 1: Automate Log File Management**

**Objective**

To automate the process of identifying, compressing, and archiving large log files while ensuring efficient disk space utilization.

**Script Overview**

The Bash script performs the following steps:

1. Searches for log files larger than a specified size in a given directory.
2. Compresses these files into a timestamped archive.
3. Moves the archive to a backup directory.
4. Deletes the original log files after compression

**Scripting Code:**

A screenshot of a computer

Description automatically generated

Save the code as: **log\_management.sh**

And use the command to run the script

A screenshot of a computer

Description automatically generated

OUTPUT:

A screenshot of a computer screen

Description automatically generated

**Task 2: Create a System Health Monitoring Script**

**Objective**

To monitor CPU and memory usage in real-time and log this data into a file every minute for a specified duration.

**Script Overview**

The script:

1. Captures CPU and memory usage using the top command.
2. Logs this data into a file at one-minute intervals for a specified duration.
3. Ensures logs are timestamped for clear and actionable insights.

**Scripting Code:**

**A screenshot of a computer

Description automatically generated**

**Save the code as:** system\_health\_monitor.sh

And use the command to run the script

A screenshot of a computer

Description automatically generated

Output:

A screenshot of a computer

Description automatically generated

**Real-World Application**

**System health monitoring scripts provide real-time performance data, enabling administrators to detect anomalies, troubleshoot issues, and ensure system stability. This data can be invaluable for capacity planning and optimizing system performance.**

**Task 3: Automate Software Installation**

**Objective**

To automate the installation of software packages by reading a list from a text file and using the system’s package manager.

**Script Overview**

The script:

1. Reads a list of software package names from a text file.
2. Installs each package using the system’s package manager (e.g., apt).
3. Logs the outcome of each installation attempt.

Scripting code :

A screenshot of a computer

Description automatically generated

Save the code as: **software\_install.sh**

And use the command to run the script

A screenshot of a computer

Description automatically generated

Output:

A computer screen with white text

Description automatically generated

**Real-World Application**

This script simplifies and standardizes the installation of software packages, making it particularly useful for provisioning new servers or setting up development environments. Automating this process ensures consistency and saves time.