# GROUP 8

| NAME | INDEX NUMBER |
|---|---|
| BISMARK TAWIAH | 052441360129 |
| AGBEKE MARK MAKAFUI | 052441360168 |
| IBRAHIM ZAINAB | 052441360300 |
| PRISCILLA SARPONG | 052442620001 |

# Automated Package Update System – Full Report

## A. Writing and Testing `update_system.sh`

The `update_system.sh` script was developed to automatically check for and apply updates to the system using the `apt` package manager. The script included the following main steps:

1. **Update package lists** using `sudo apt update` to fetch the latest package information.
2. **Upgrade packages** using `sudo apt upgrade -y` to automatically install available updates.
3. **Log results** to a specific log file to keep track of when updates were performed and what was changed.
4. **Error handling** was added to ensure the script stopped and reported any issues during updates.

**Testing Process:**

- The script was tested by running it manually with `bash update_system.sh`.
- The output confirmed that the update and upgrade processes completed successfully.
- The generated log file included timestamps and detailed package update information, confirming the script worked as intended.

---

## B. Scheduling the Script with Cron

To ensure the updates happen automatically, the script was scheduled to run **weekly** using a cron job. This was done by:

1. Opening the cron editor with `crontab -e`.
2. Adding an entry such as:

```
3. 0 9 * * 1 /path/to/update_system.sh
```

This ensures the script runs every Monday at 9:00 AM.

4. Testing the cron job by temporarily setting a shorter interval confirmed that it executed the script without manual intervention.

---

## C. Skills and Concepts Used

This project involved skills from three key areas:

1. **Package Management (Week 3):**
    - Learned and applied the use of `apt update` and `apt upgrade` for managing software packages.
    - Understood the importance of running `apt update` before `apt upgrade` to ensure the system installs the latest versions.
2. **Automation (Week 4):**
    - Used cron jobs to schedule automated tasks.
    - Learned cron syntax and how to check scheduled jobs with `crontab -l`.
3. **Scripting (Week 6):**
    - Wrote a Bash script with proper syntax, logging, and error handling.
    - Used shell commands like `date` for timestamping logs and `tee` for writing output to both the terminal and log files.

---

## D. Security Considerations

- The script was run with `sudo` privileges only when necessary to reduce security risks.
- Automatic updates help reduce vulnerabilities by ensuring the system always runs the latest security patches.
- Logs were stored in a secure directory to avoid unauthorized access.

---

## E. Conclusion

The **Automated Package Update System** successfully ensures that the system stays up to date without requiring manual intervention. The combination of Bash scripting, package management commands, and cron job automation results in a reliable solution for maintaining system security and stability.

This approach can be further improved by:

- Adding email notifications for update results.
- Implementing a check for kernel updates and scheduling reboots if necessary.

- Expanding compatibility to other package managers like `dnf` or `yum` for use on non-Ubuntu systems.