Linux Command Line Assignment

Objective:

Develop hands-on skills with the Linux command line by completing tasks involving comprehensive file management, content manipulation, system monitoring, network testing, and scripting, along with systematic documentation of the process.

Tasks – Make sure to document every step (can be screenshots):

1. **Version Control Initialization and Repository Setup**
    - Create a GitHub repository `robotics-assignment-2`.
    - Clone the repository locally and navigate into it.
    - Commit each step with the necessary details (don't just push the whole project), and use the commit conventions mentioned previously.

2. **Advanced Directory and File Management**
    - Use `mkdir` to create a structured directory: `workspace/{docs,logs,data,scripts}`.
    - Navigate to `workspace` and create initial setup files using `touch`.
    - Take screenshots of the directory structure and initial file setup.

3. **File Creation and Manipulation**
    - **Steps** (Document each command's use and output with screenshots):
        1. Use `touch` to create several files of your choice within `docs` and fill them up with some content using `nano`.
        2. Utilize `echo` to add introductory content to a file called `welcome.txt`.
        3. Merge multiple text files into one using `cat` and redirect the output to `summary.txt`.
        4. Display the beginning of `summary.txt` with `head` and the end with `tail`.
        5. Use `grep` to find specific strings in `summary.txt` and redirect these findings to `logs/search_results.txt`.
        6. Move and copy files within the project directories using `mv` and `cp`.

4. **File Permissions and Ownership**
    - Adjust permissions of `scripts` to allow only the user execution rights.
    - Change ownership of `data` to the student and a group (names of your choice).
    - Use `ls -l` to check the rights.

5. **System Monitoring and Basic Networking**
    - Monitor disk space with `df -h` and directory size with `du`.

- Display network configurations with `ifconfig`.
- Check internet connectivity using `ping`.
- Capture and document all outputs.

6. **Bash Script for Automation**
    - Write a Bash script `setup.sh` in the `scripts` folder to automate the directory and file setup, including setting permissions.
    - Include explanatory comments in the script.
    - Execute the script, screenshot the process, and the results.
    - When this script runs on any machine, we should get the same structure for the project you have with all the details.

7. **Documentation and Reporting**
    - Compile all tasks, screenshots, and command explanations into a comprehensive report.
    - Include the GitHub repository link showcasing the entire project structure and scripts.

Deliverables:

- A detailed PDF report containing a visualization of each task, using screenshots of the terminal or the contents of folders/files (use whatever makes more sense to you).
- A GitHub repository link with all project files and documentation.