backup.sh

```
#!/bin/bash

# Create backup folder if not already there
mkdir -p backup

# Get current date and time for timestamp
timestamp=$(date +"%Y%m%d_%H%M%S")

# Loop to find all .txt files and copy them with timestamp
for file in *.txt; do
  if [ -f "$file" ]; then
    cp "$file" "backup/${file%.txt}_$timestamp.txt"
  fi
done

echo "Backup completed successfully."
```

LAB5.md

```
# LAB5 – File & Backup Automation

---

## Objective

To write a bash script that copies all `.txt` files in the current folder into a
backup folder with a timestamp so files are not overwritten. This helps to
automate file backup.

---

## How the Script Works

1. The script first creates a folder named `backup` if it does not already exist
using `mkdir -p backup`.
2. It gets the current date and time using the `date` command and saves it as a
timestamp.
3. Then it searches for all `.txt` files in the current directory using a `for`
loop.
4. For each `.txt` file found, it copies the file into the `backup` folder and
adds the timestamp at the end of the filename.
5. Finally, it prints a message “Backup completed successfully.” after all files
are copied.

---
```

~~~
## Example Run

1. I created some sample `.txt` files for testing:

```bash
echo "Hello world" > file1.txt
echo "This is a test" > file2.txt
~~~

2. Ran the script:

```
./backup.sh
```

3. Output:

```
Backup completed successfully.
```

4. Checked the backup folder:

```
backup/file1_20250909_153000.txt
backup/file2_20250909_153000.txt
```

Files copied successfully with timestamp added.

Extra Questions

Q1: What is the difference between `cp`, `mv`, and `rsync`?

- `cp` copies files or folders from one place to another, keeping the original file.
- `mv` moves or renames files/folders (the original file is removed from the source).
- `rsync` is used to sync files between locations efficiently, copying only the changes and useful for backups.

Q2: How can you schedule scripts to run automatically?

We can use **cron jobs** in Linux to schedule scripts.

- Open terminal and type `crontab -e`.
- Add a line to schedule the script. For example, to run every day at 7 AM:

```
0 7 * * * /path/to/backup.sh
```

This runs the backup script automatically every day at 7 in the morning.

Conclusion

This assignment helped me learn how to automate file backups using bash scripting. I learned to use loops, conditions, timestamps, and basic file operations in Linux.

Deliverables

- `backup.sh`
- `LAB5.md`
- `LAB5.pdf` (exported from this markdown)