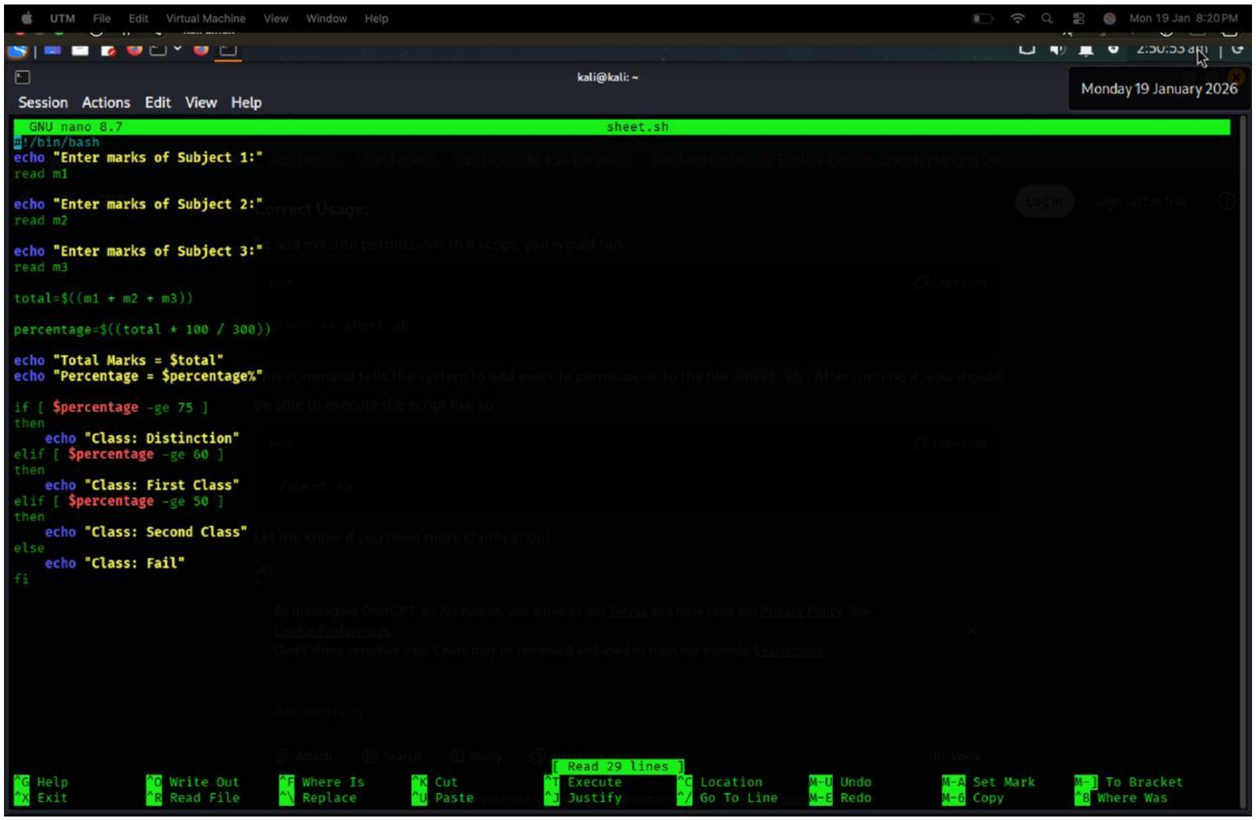
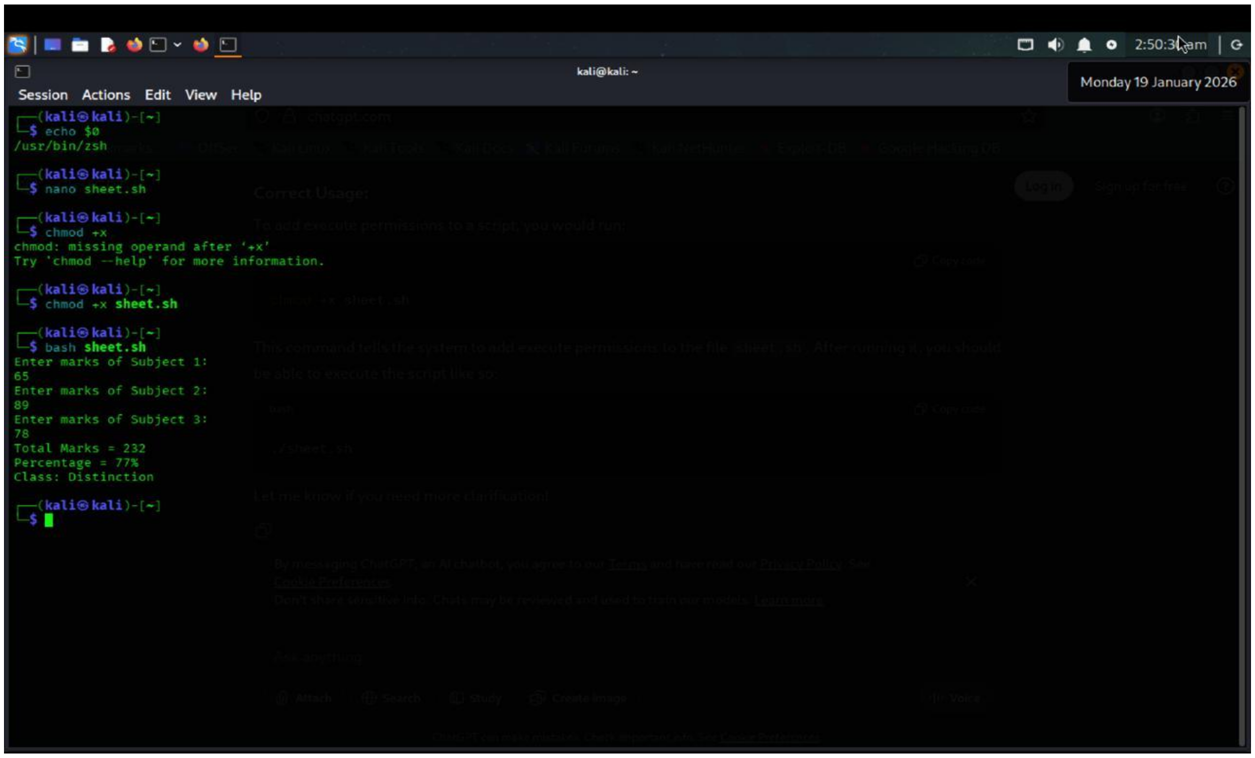
1.Marksheet Genration



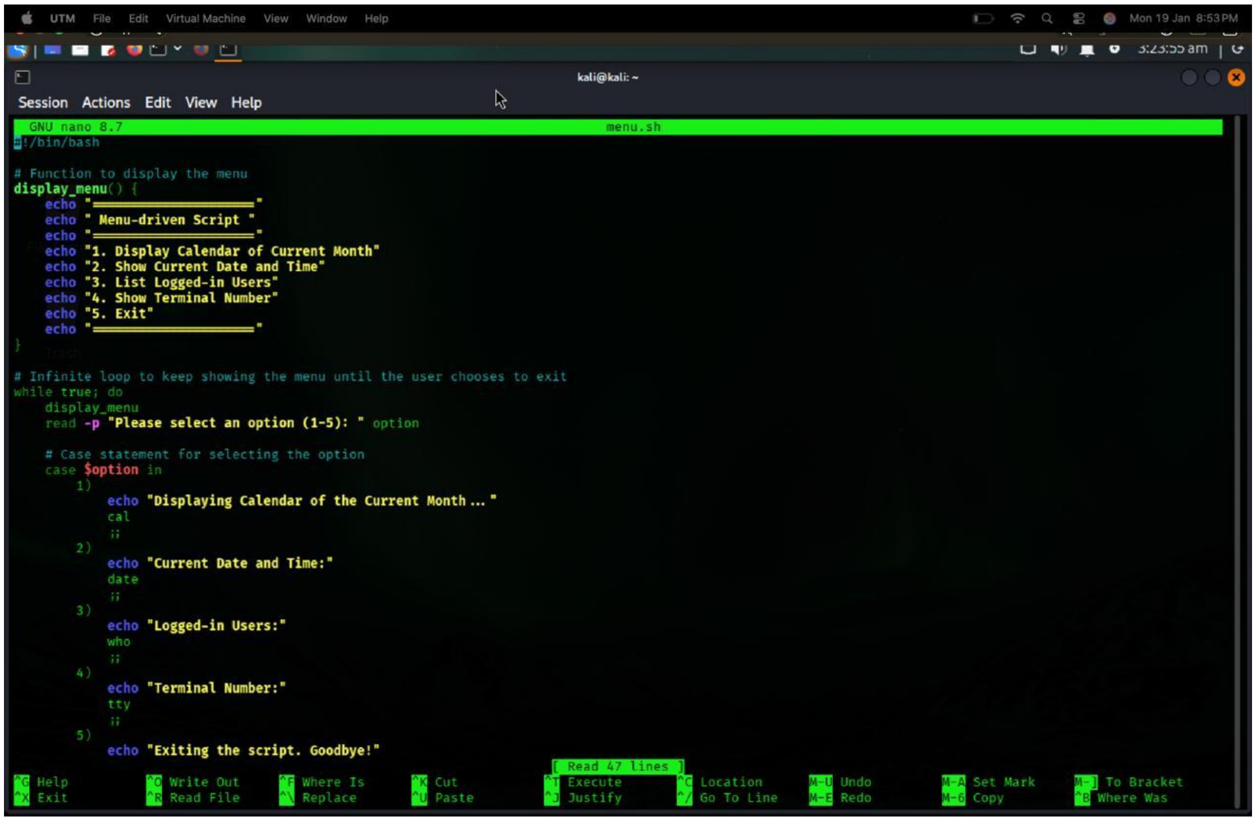
```
#!/bin/bash
echo "Enter marks of Subject 1:"
read m1

echo "Enter marks of Subject 2:"
read m2

echo "Enter marks of Subject 3:"
read m3

total=$((m1 + m2 + m3))

percentage=$((total * 100 / 300))

echo "Total Marks = $total"
echo "Percentage = $percentage%"

if [ $percentage -ge 75 ]
then
    echo "Class: Distinction"
elif [ $percentage -ge 60 ]
then
    echo "Class: First Class"
elif [ $percentage -ge 50 ]
then
    echo "Class: Second Class"
else
    echo "Class: Fail"
fi
```

Output:-

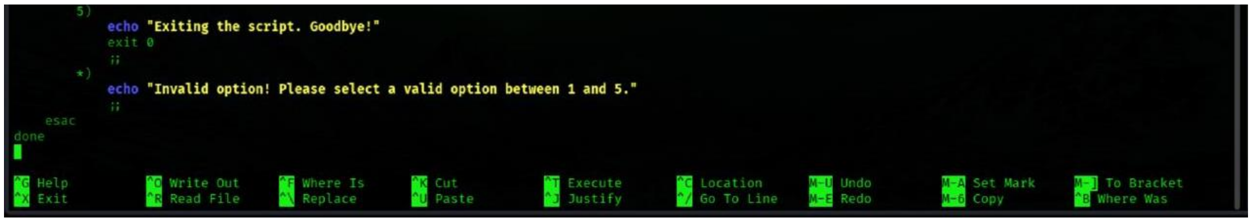


```
┌──(kali㉿kali)-[~]
└─$ echo $0
/usr/bin/zsh

┌──(kali㉿kali)-[~]
└─$ nano sheet.sh

┌──(kali㉿kali)-[~]
└─$ chmod +x
chmod: missing operand after '+x'
Try 'chmod --help' for more information.

┌──(kali㉿kali)-[~]
└─$ chmod +x sheet.sh

┌──(kali㉿kali)-[~]
└─$ bash sheet.sh
Enter marks of Subject 1:
65
Enter marks of Subject 2:
89
Enter marks of Subject 3:
78
Total Marks = 232
Percentage = 77%
Class: Distinction

┌──(kali㉿kali)-[~]
└─$ 
```

2. Menu-Driven Script for System Information.

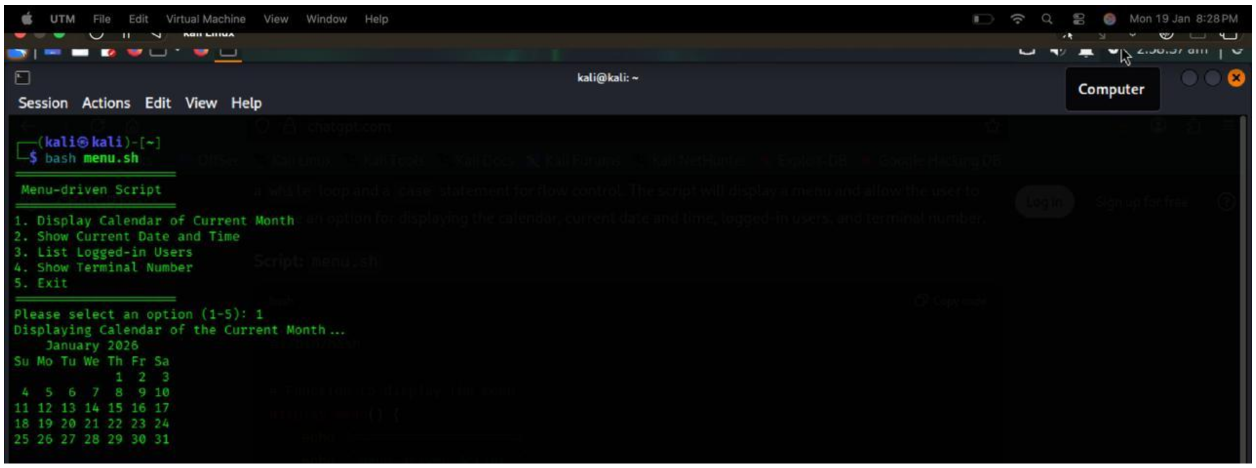


```
#!/bin/bash

# Function to display the menu
display_menu() {
    echo "======================="
    echo " Menu-driven Script "
    echo "======================="
    echo "1. Display Calendar of Current Month"
    echo "2. Show Current Date and Time"
    echo "3. List Logged-in Users"
    echo "4. Show Terminal Number"
    echo "5. Exit"
    echo "======================="
}

# Infinite loop to keep showing the menu until the user chooses to exit
while true; do
    display_menu
    read -p "Please select an option (1-5): " option

    # Case statement for selecting the option
    case $option in
        1)
            echo "Displaying Calendar of the Current Month..."
            cal
            ;;
        2)
            echo "Current Date and Time:"
            date
            ;;
        3)
            echo "Logged-in Users:"
            who
            ;;
        4)
            echo "Terminal Number:"
            tty
            ;;
        5)
            echo "Exiting the script. Goodbye!"
```

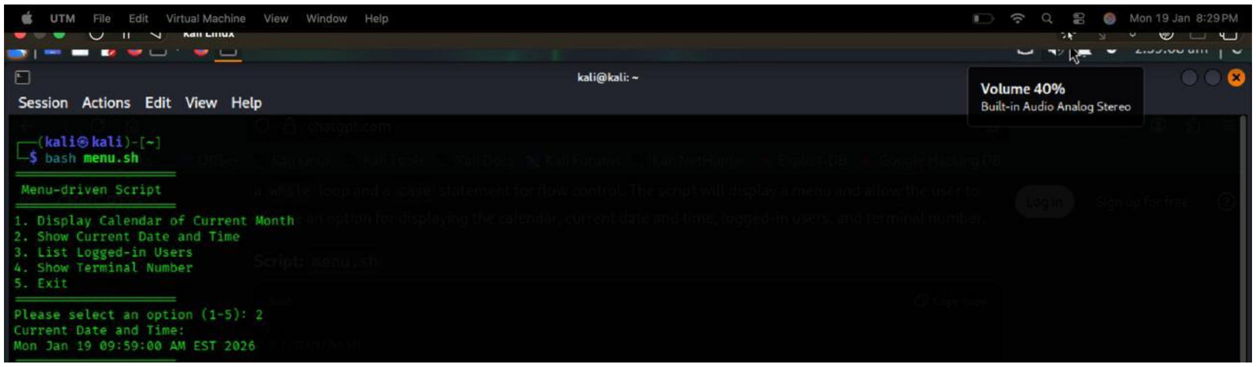


```
        5)
            echo "Exiting the script. Goodbye!"
            exit 0
            ;;
        *)
            echo "Invalid option! Please select a valid option between 1 and 5."
            ;;
    esac
done
```

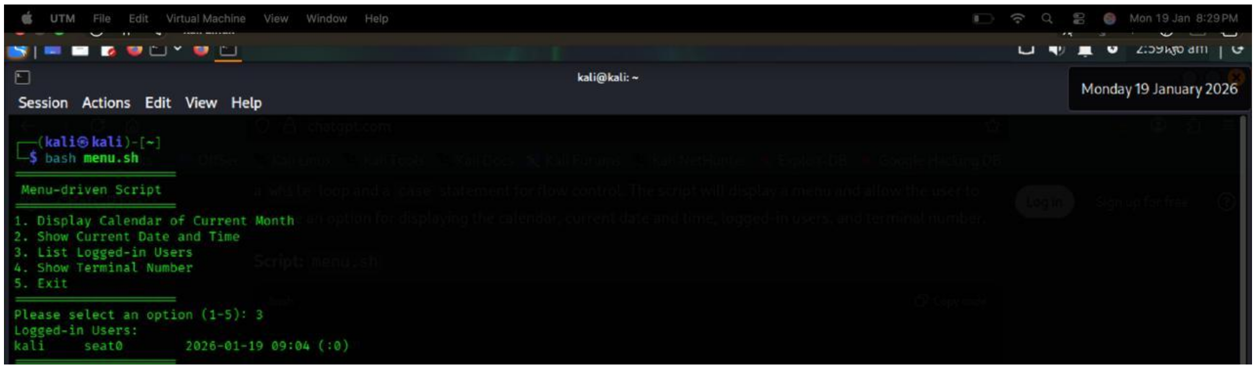
Output:-



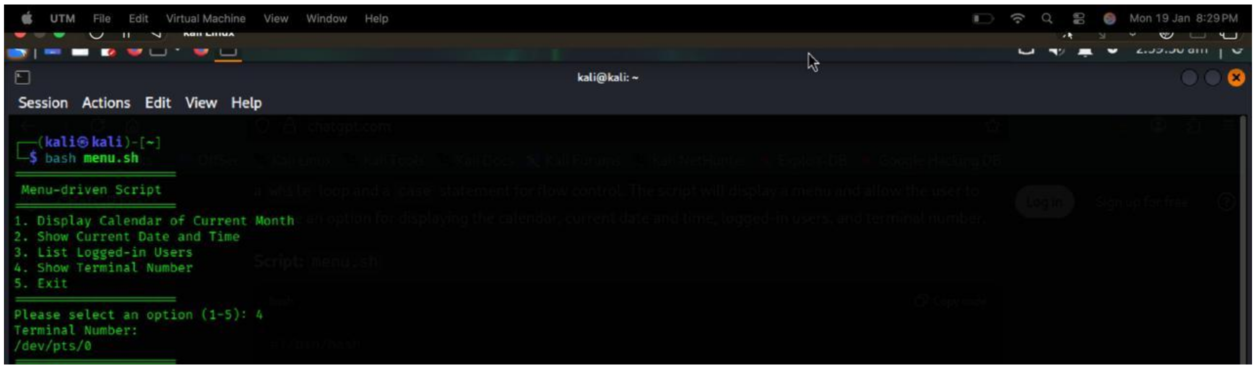
```
┌──(kali㉿kali)-[~]
└─$ bash menu.sh
=======================
 Menu-driven Script
=======================
1. Display Calendar of Current Month
2. Show Current Date and Time
3. List Logged-in Users
4. Show Terminal Number
5. Exit
=======================
Please select an option (1-5): 1
Displaying Calendar of the Current Month...
    January 2026
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```



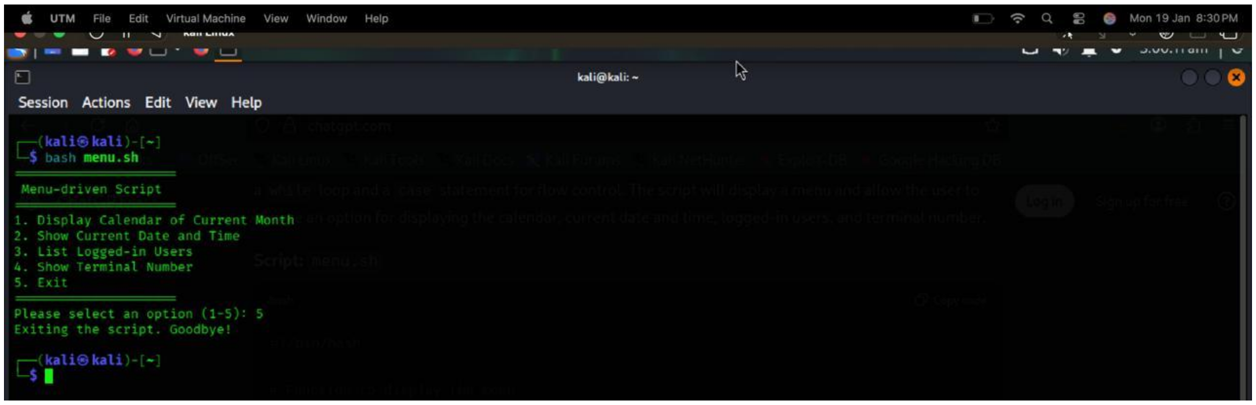
```
┌──(kali㉿kali)-[~]
└─$ bash menu.sh
=========================
 Menu-driven Script
=========================
1. Display Calendar of Current Month
2. Show Current Date and Time
3. List Logged-in Users
4. Show Terminal Number
5. Exit
=========================
Please select an option (1-5): 2
Current Date and Time:
Mon Jan 19 09:59:00 AM EST 2026
```



```
┌──(kali㉿kali)-[~]
└─$ bash menu.sh
=========================
 Menu-driven Script
=========================
1. Display Calendar of Current Month
2. Show Current Date and Time
3. List Logged-in Users
4. Show Terminal Number
5. Exit
=========================
Please select an option (1-5): 3
Logged-in Users:
kali     seat0        2026-01-19 09:04 (:0)
```

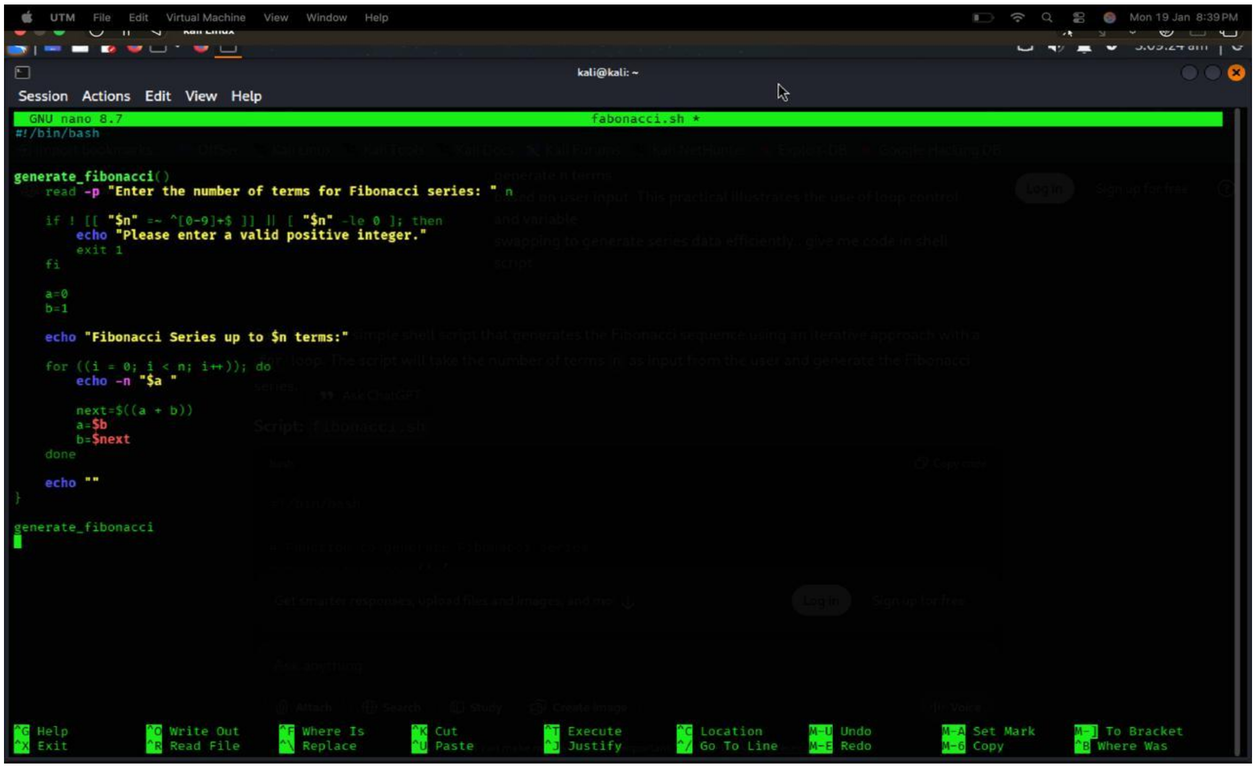


```
┌──(kali㉿kali)-[~]
└─$ bash menu.sh
=========================
 Menu-driven Script
=========================
1. Display Calendar of Current Month
2. Show Current Date and Time
3. List Logged-in Users
4. Show Terminal Number
5. Exit
=========================
Please select an option (1-5): 4
Terminal Number:
/dev/pts/0
=========================
```



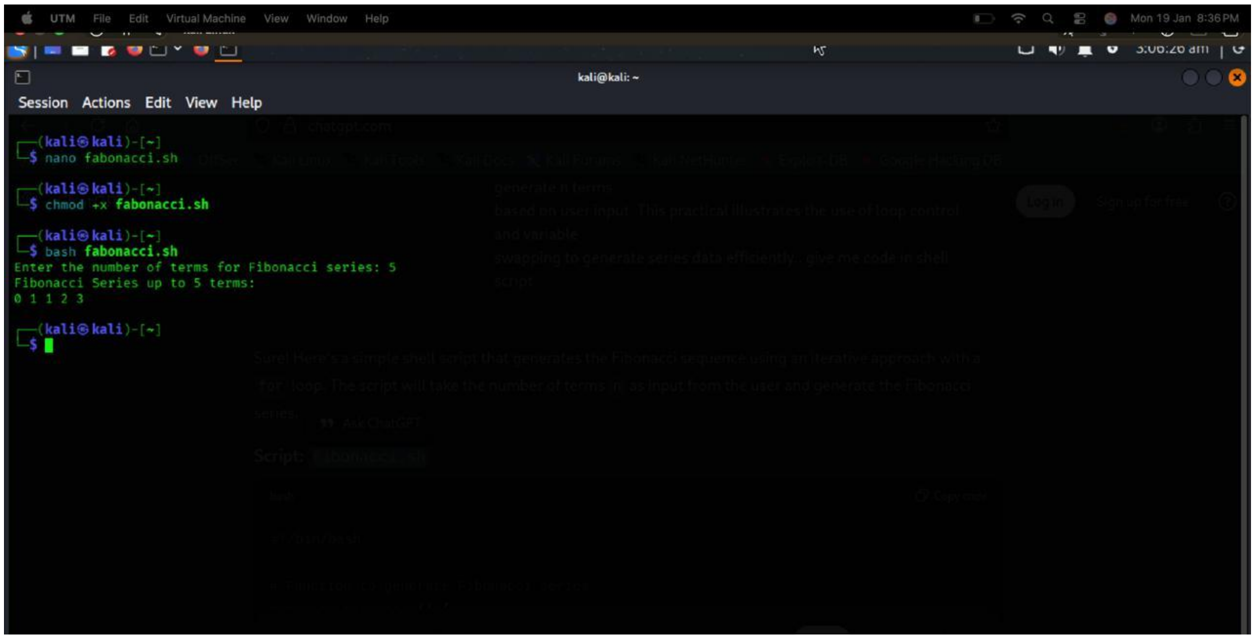
```
┌──(kali㉿kali)-[~]
└─$ bash menu.sh
=========================
 Menu-driven Script
=========================
1. Display Calendar of Current Month
2. Show Current Date and Time
3. List Logged-in Users
4. Show Terminal Number
5. Exit
=========================
Please select an option (1-5): 5
Exiting the script. Goodbye!

┌──(kali㉿kali)-[~]
└─$ 
```

3. Fibonacci Number Generation.



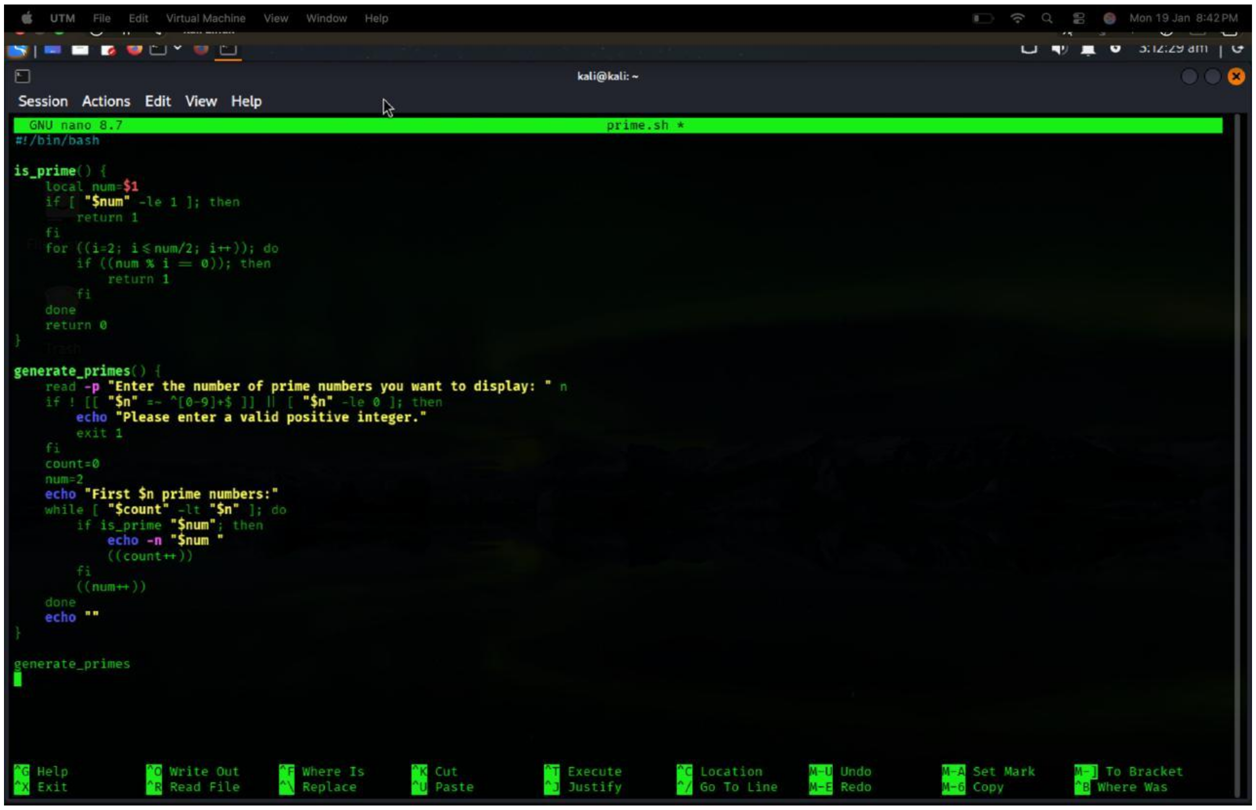
```
GNU nano 8.7                          fabonacci.sh *
#!/bin/bash

generate_fibonacci()
    read -p "Enter the number of terms for Fibonacci series: " n

    if ! [[ "$n" =~ ^[0-9]+$ ]] || [ "$n" -le 0 ]; then
        echo "Please enter a valid positive integer."
        exit 1
    fi

    a=0
    b=1

    echo "Fibonacci Series up to $n terms:"

    for ((i = 0; i < n; i++)); do
        echo -n "$a "

        next=$((a + b))
        a=$b
        b=$next
    done

    echo ""
}

generate_fibonacci
```

Output:-



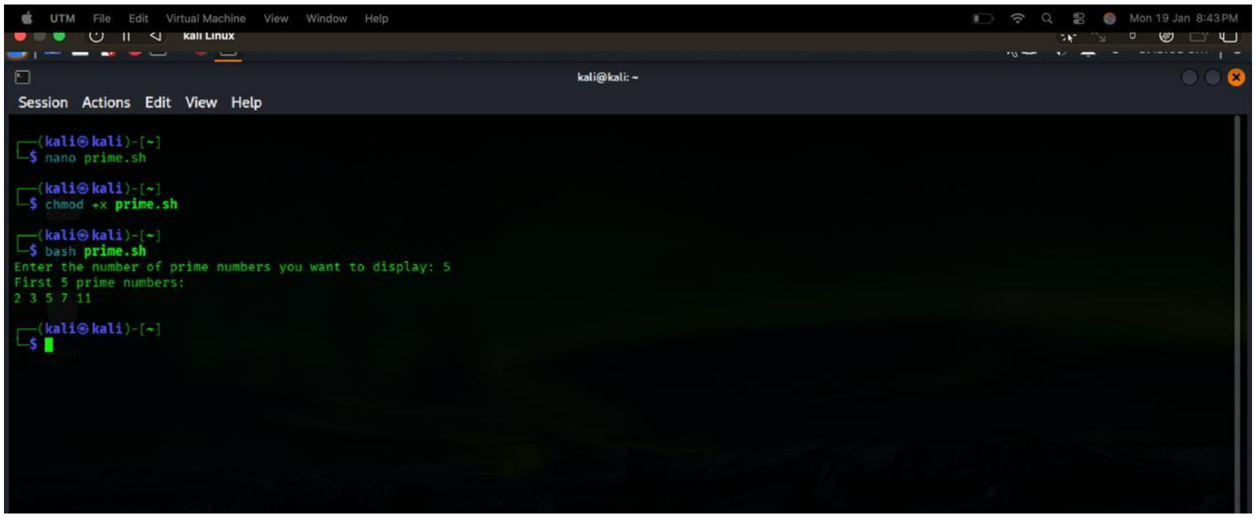
```
┌──(kali㉿kali)-[~]
└─$ nano fabonacci.sh

┌──(kali㉿kali)-[~]
└─$ chmod +x fabonacci.sh

┌──(kali㉿kali)-[~]
└─$ bash fabonacci.sh
Enter the number of terms for Fibonacci series: 5
Fibonacci Series up to 5 terms:
0 1 1 2 3

┌──(kali㉿kali)-[~]
└─$ 
```

4. Prime Number Display.



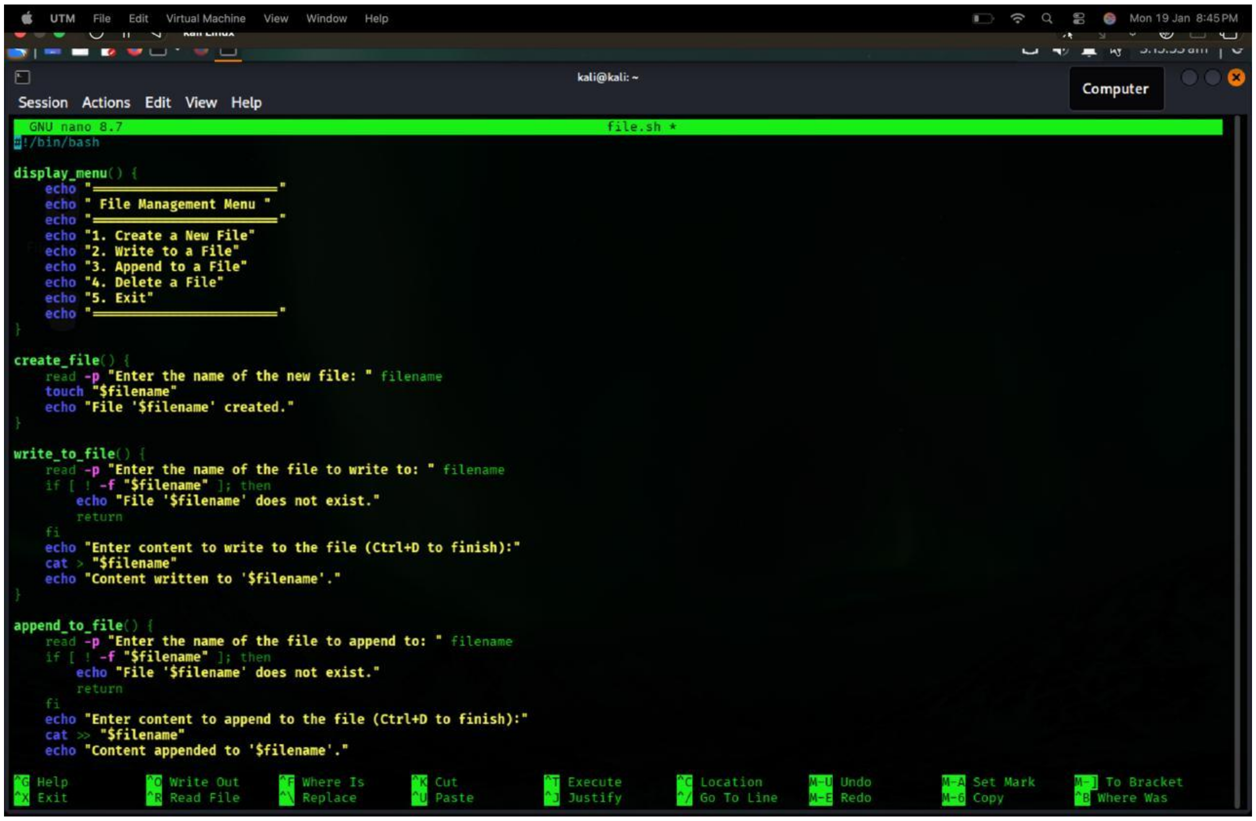
```
GNU nano 8.7                                   prime.sh *
#!/bin/bash

is_prime() {
    local num=$1
    if [ "$num" -le 1 ]; then
        return 1
    fi
    for ((i=2; i<=num/2; i++)); do
        if ((num % i == 0)); then
            return 1
        fi
    done
    return 0
}

generate_primes() {
    read -p "Enter the number of prime numbers you want to display: " n
    if ! [[ "$n" =~ ^[0-9]+$ ]] || [ "$n" -le 0 ]; then
        echo "Please enter a valid positive integer."
        exit 1
    fi
    count=0
    num=2
    echo "First $n prime numbers:"
    while [ "$count" -lt "$n" ]; do
        if is_prime "$num"; then
            echo -n "$num "
            ((count++))
        fi
        ((num++))
    done
    echo ""
}

generate_primes
```

Output:-

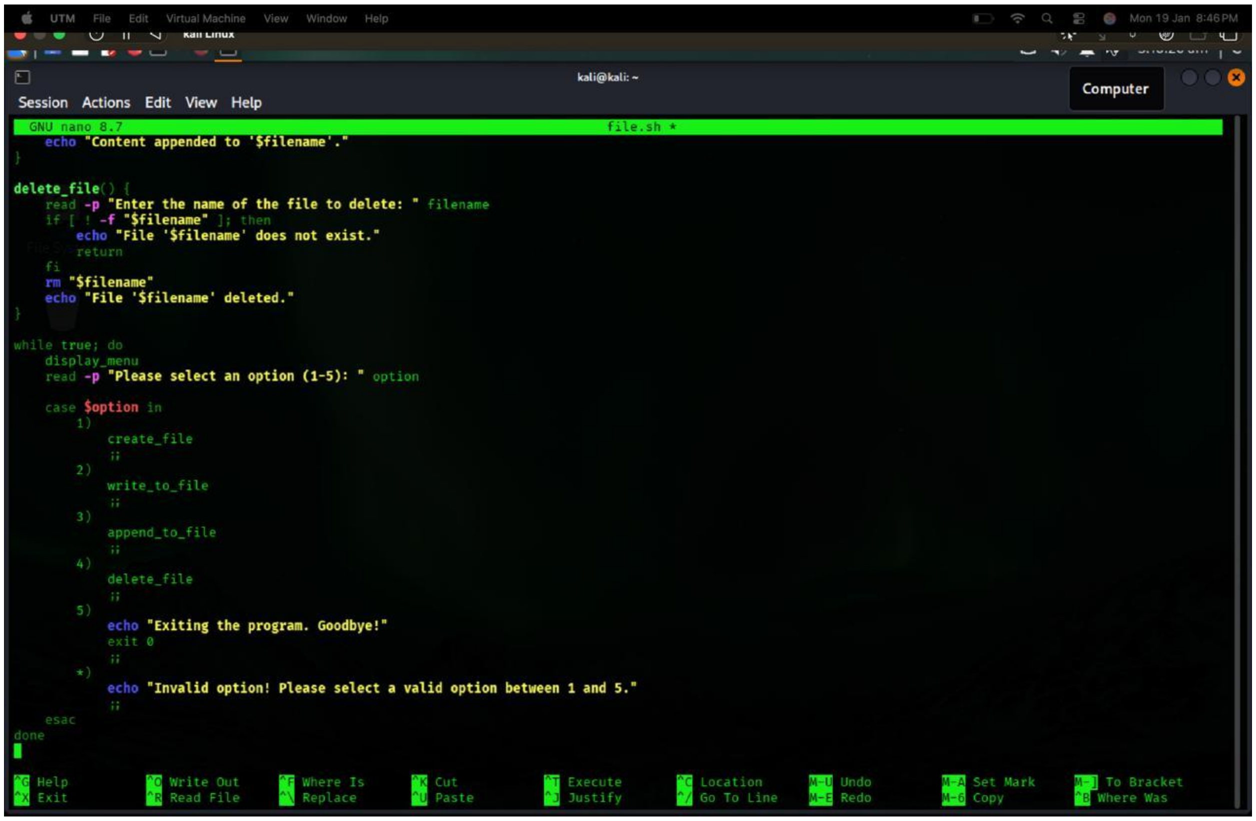


```
┌──(kali㉿kali)-[~]
└─$ nano prime.sh

┌──(kali㉿kali)-[~]
└─$ chmod +x prime.sh

┌──(kali㉿kali)-[~]
└─$ bash prime.sh
Enter the number of prime numbers you want to display: 5
First 5 prime numbers:
2 3 5 7 11

┌──(kali㉿kali)-[~]
└─$
```

5. Menu-Driven File Management.

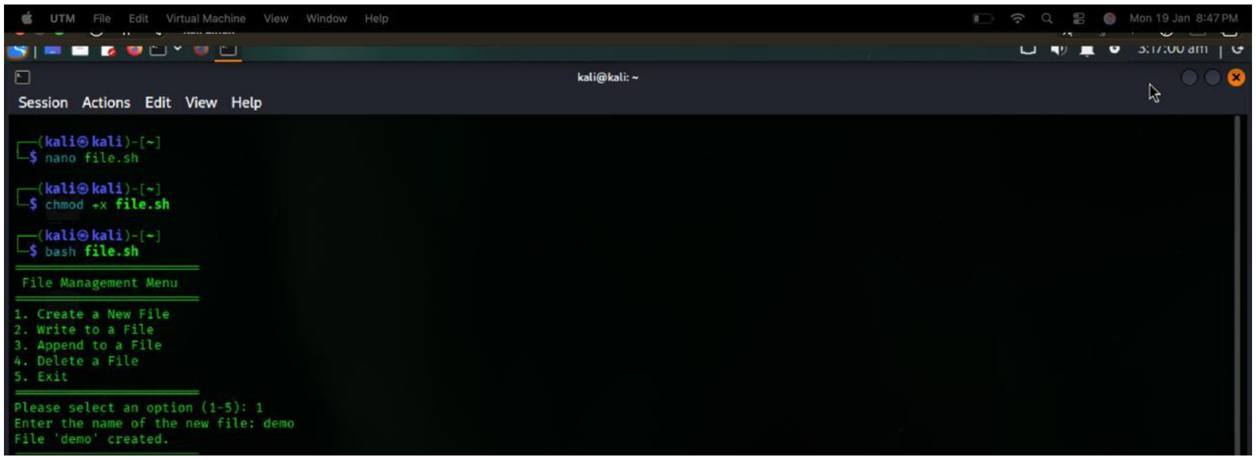

UTM File Edit Virtual Machine View Window Help
Mon 19 Jan 8:45 PM
kali@kali: ~
Computer
Session Actions Edit View Help

```
  GNU nano 8.7                          file.sh *
#!/bin/bash

display_menu() {
    echo "========================="
    echo " File Management Menu "
    echo "========================="
    echo "1. Create a New File"
    echo "2. Write to a File"
    echo "3. Append to a File"
    echo "4. Delete a File"
    echo "5. Exit"
    echo "========================="
}

create_file() {
    read -p "Enter the name of the new file: " filename
    touch "$filename"
    echo "File '$filename' created."
}

write_to_file() {
    read -p "Enter the name of the file to write to: " filename
    if [ ! -f "$filename" ]; then
        echo "File '$filename' does not exist."
        return
    fi
    echo "Enter content to write to the file (Ctrl+D to finish):"
    cat > "$filename"
    echo "Content written to '$filename'."
}

append_to_file() {
    read -p "Enter the name of the file to append to: " filename
    if [ ! -f "$filename" ]; then
        echo "File '$filename' does not exist."
        return
    fi
    echo "Enter content to append to the file (Ctrl+D to finish):"
    cat >> "$filename"
    echo "Content appended to '$filename'."
^G Help       ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^B Where Was
```

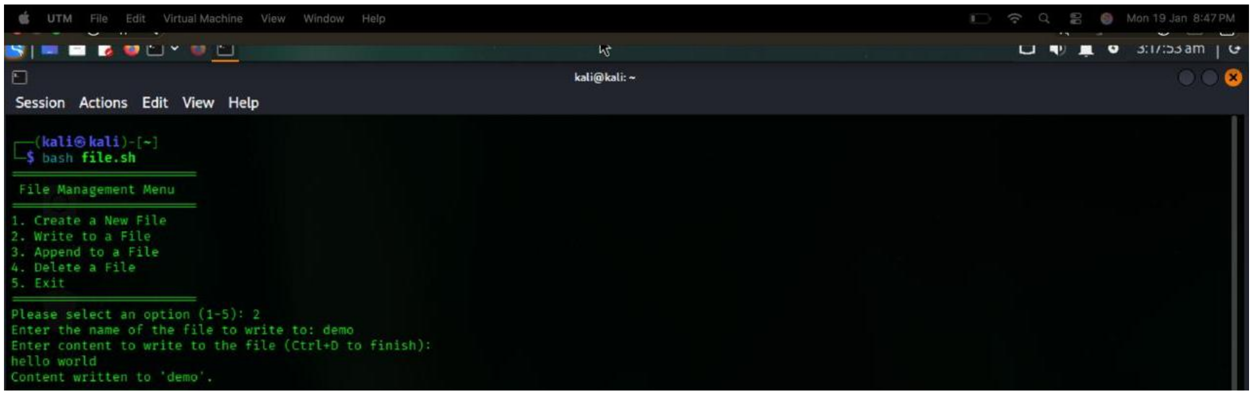

UTM File Edit Virtual Machine View Window Help
Mon 19 Jan 8:46 PM
kali@kali: ~
Computer
Session Actions Edit View Help

```
  GNU nano 8.7                          file.sh *
    echo "Content appended to '$filename'."
}

delete_file() {
    read -p "Enter the name of the file to delete: " filename
    if [ ! -f "$filename" ]; then
        echo "File '$filename' does not exist."
        return
    fi
    rm "$filename"
    echo "File '$filename' deleted."
}

while true; do
    display_menu
    read -p "Please select an option (1-5): " option

    case $option in
        1)
            create_file
            ;;
        2)
            write_to_file
            ;;
        3)
            append_to_file
            ;;
        4)
            delete_file
            ;;
        5)
            echo "Exiting the program. Goodbye!"
            exit 0
            ;;
        *)
            echo "Invalid option! Please select a valid option between 1 and 5."
            ;;
    esac
done

^G Help       ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^B Where Was
```

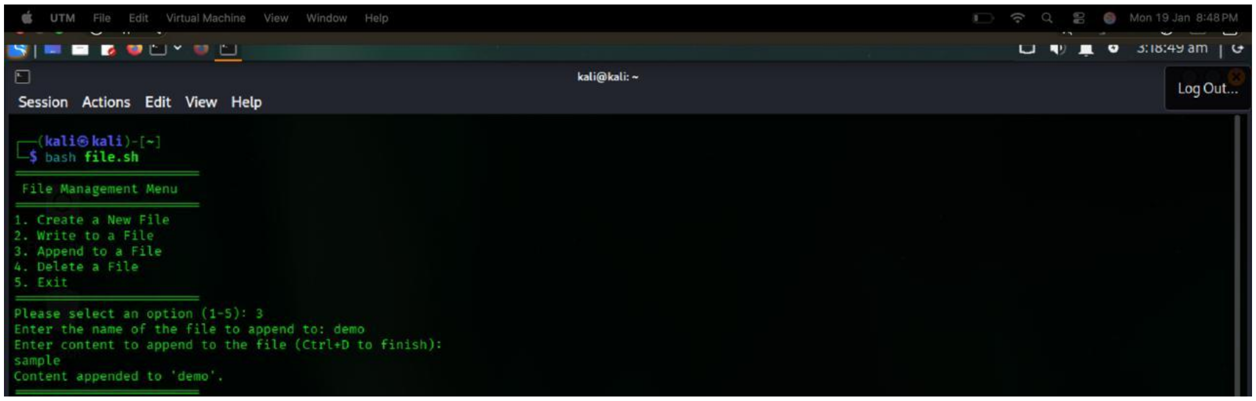
Output:-



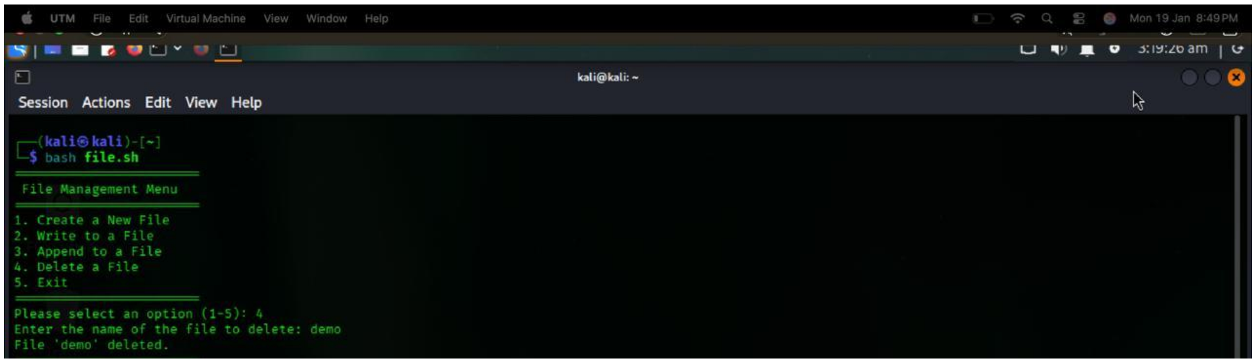
```
┌──(kali㉿kali)-[~]
└─$ nano file.sh

┌──(kali㉿kali)-[~]
└─$ chmod +x file.sh

┌──(kali㉿kali)-[~]
└─$ bash file.sh
========================
 File Management Menu
========================
1. Create a New File
2. Write to a File
3. Append to a File
4. Delete a File
5. Exit
========================
Please select an option (1-5): 1
Enter the name of the new file: demo
File 'demo' created.
```



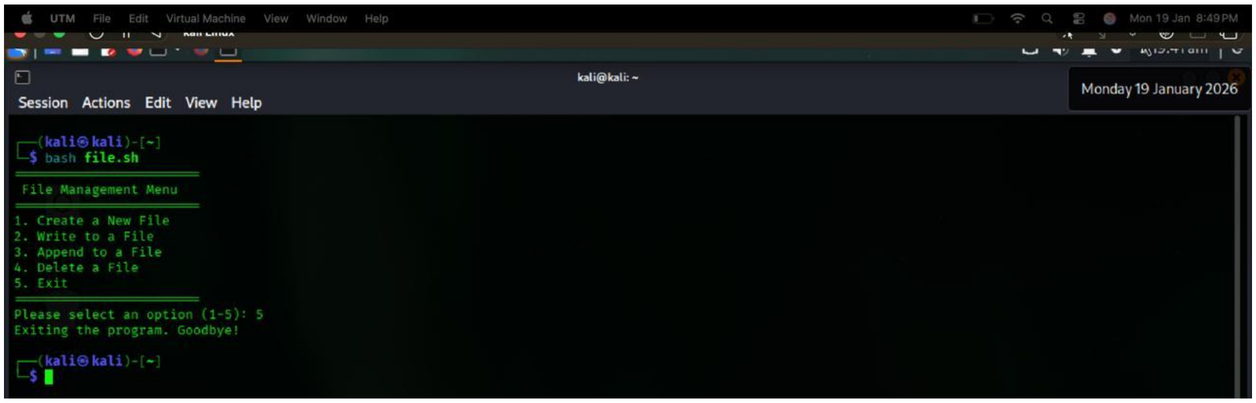
```
┌──(kali㉿kali)-[~]
└─$ bash file.sh
========================
 File Management Menu
========================
1. Create a New File
2. Write to a File
3. Append to a File
4. Delete a File
5. Exit
========================
Please select an option (1-5): 2
Enter the name of the file to write to: demo
Enter content to write to the file (Ctrl+D to finish):
hello world
Content written to 'demo'.
```



```
┌──(kali㉿kali)-[~]
└─$ bash file.sh
========================
 File Management Menu
========================
1. Create a New File
2. Write to a File
3. Append to a File
4. Delete a File
5. Exit
========================
Please select an option (1-5): 3
Enter the name of the file to append to: demo
Enter content to append to the file (Ctrl+D to finish):
sample
Content appended to 'demo'.
```



```
┌──(kali㉿kali)-[~]
└─$ bash file.sh
=========================
 File Management Menu
=========================
1. Create a New File
2. Write to a File
3. Append to a File
4. Delete a File
5. Exit
=========================
Please select an option (1-5): 4
Enter the name of the file to delete: demo
File 'demo' deleted.
```



```
┌──(kali㉿kali)-[~]
└─$ bash file.sh
=========================
 File Management Menu
=========================
1. Create a New File
2. Write to a File
3. Append to a File
4. Delete a File
5. Exit
=========================
Please select an option (1-5): 5
Exiting the program. Goodbye!

┌──(kali㉿kali)-[~]
└─$ 
```