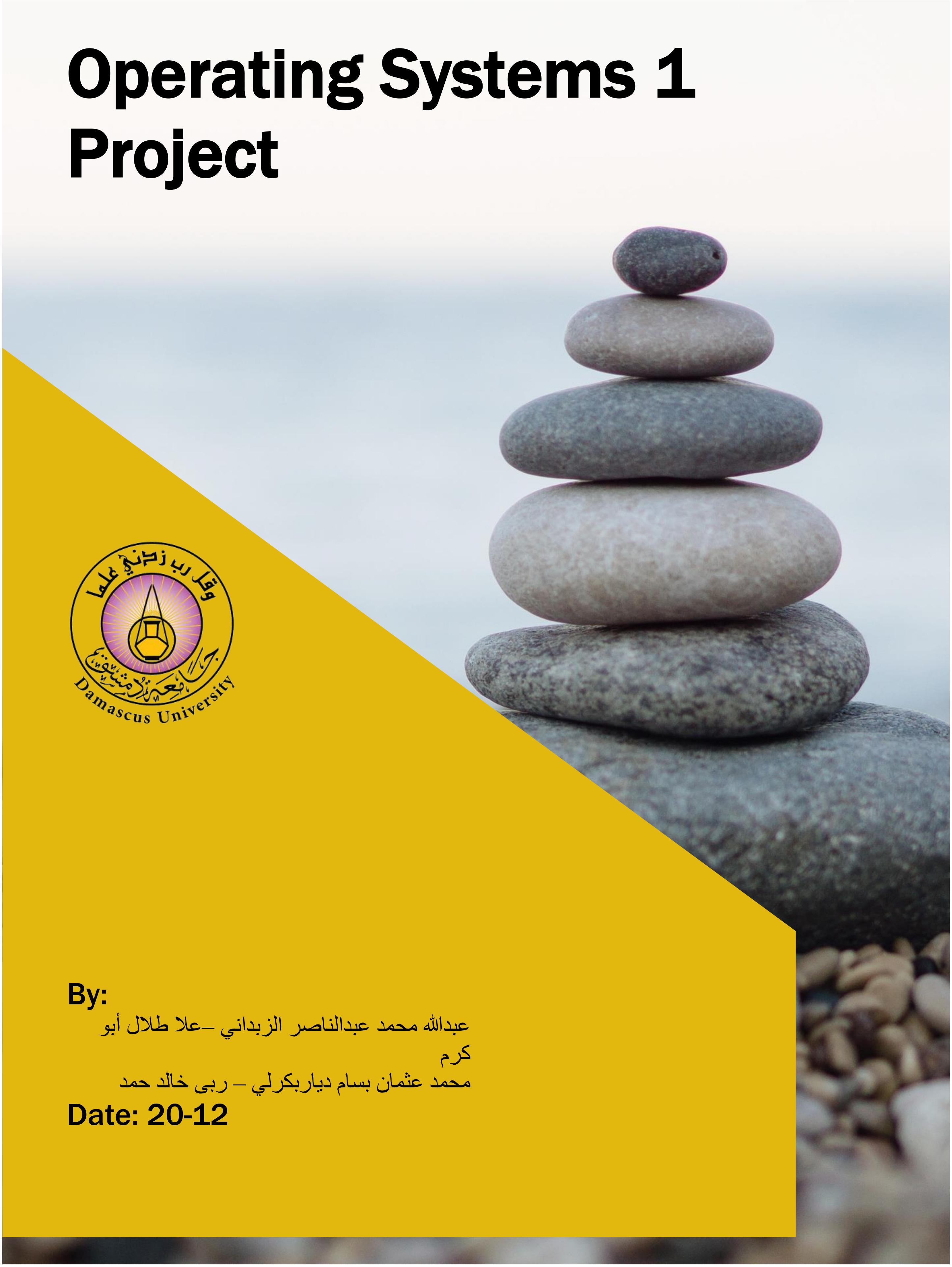
Operating Systems 1 Project



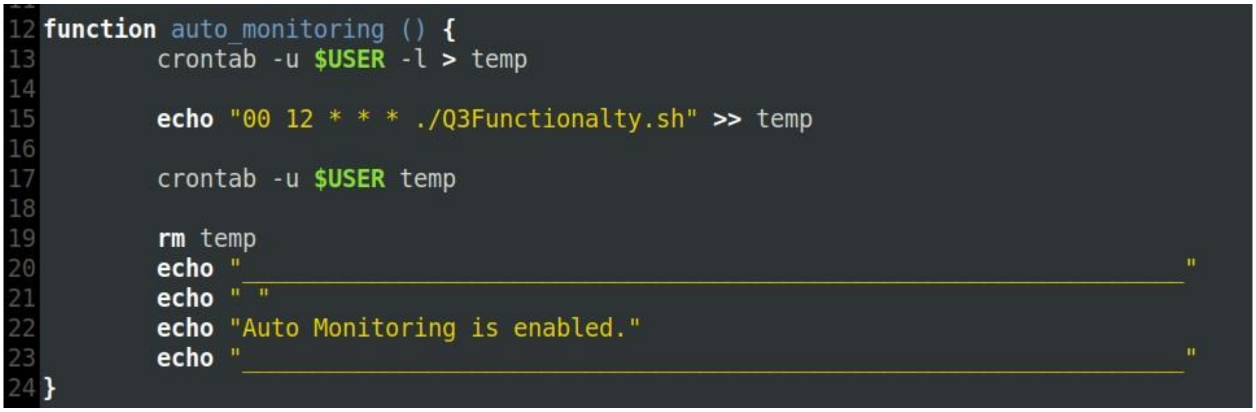
By:

عبدالله محمد عبدالناصر الزبداني –علا طلال أبو
كرم
محمد عثمان بسام دياربكرلي – ربى خالد حمد

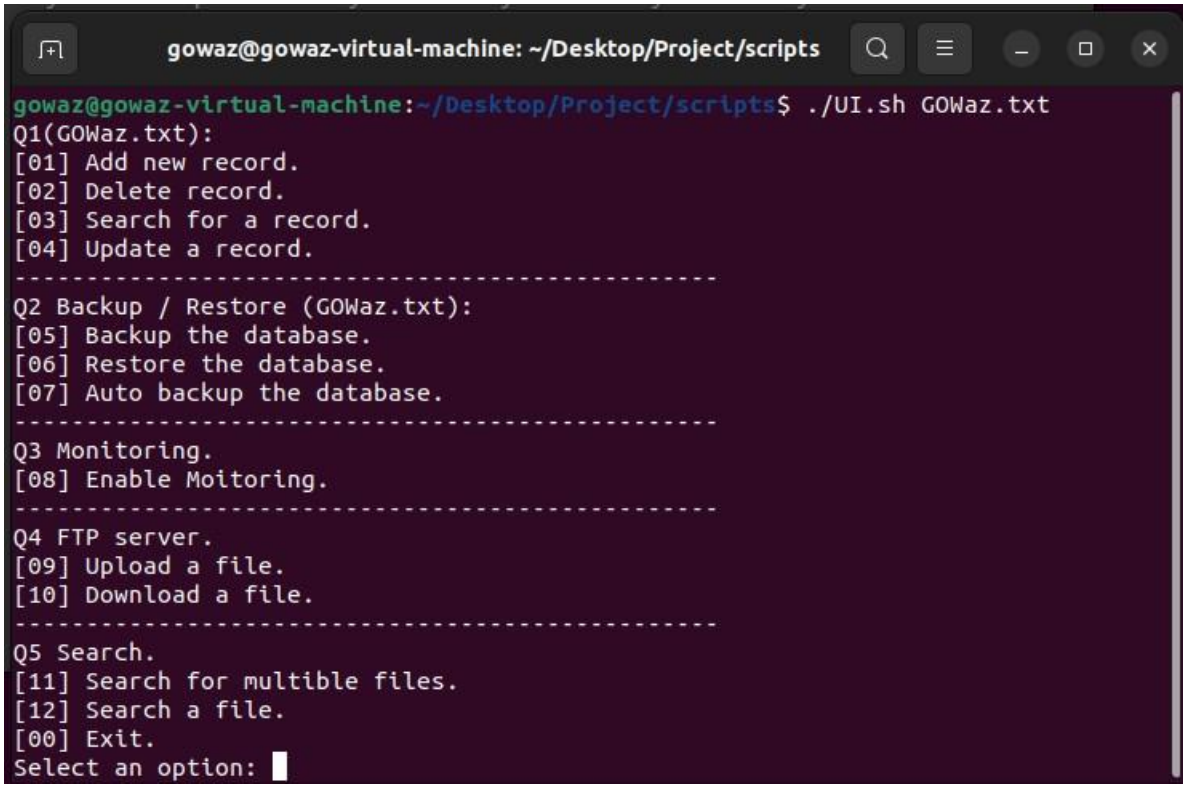
Date: 20-12

UI:

We created one script called UI.sh that runs the whole other scripts. It contains a display for the options that the user will choose from and depending on the option that the user uses it'll run either a function from another script or a whole other script, also it contains a function for the crontab for auto-monitoring for the system.



```
function auto_monitoring () {
        crontab -u $USER -l > temp

        echo "00 12 * * * ./Q3Functionalty.sh" >> temp

        crontab -u $USER temp

        rm temp
        echo "__________________________________________________________________"
        echo " "
        echo "Auto Monitoring is enabled."
        echo "__________________________________________________________________"
}
```



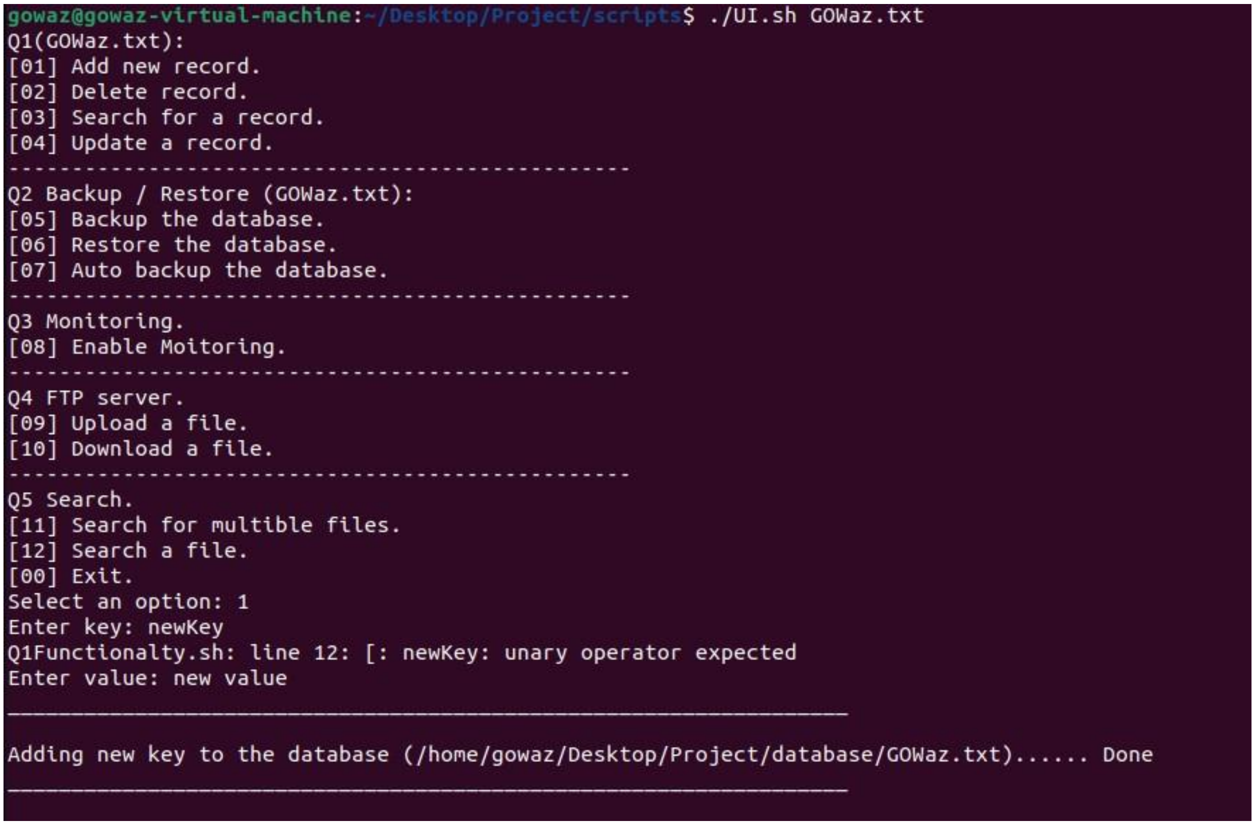
```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 
```

Q1:

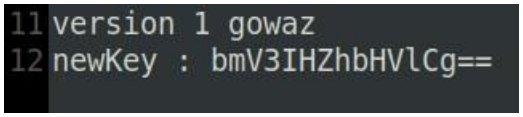
What we did here is we created one script called Q1Functionalty.sh it contains four functions each function does one of the requirements for question 1:

i. Insert new record:

First, the function asks the user to enter the key of the new record, then the function checks if the key is already in the database or not, if it's already there it displays a message "already exists." Because the key must be unique, if it doesn't exist it asks the user for the value, then it encodes the value and inserts both the key and the value in the database along with the version of the record and the user.



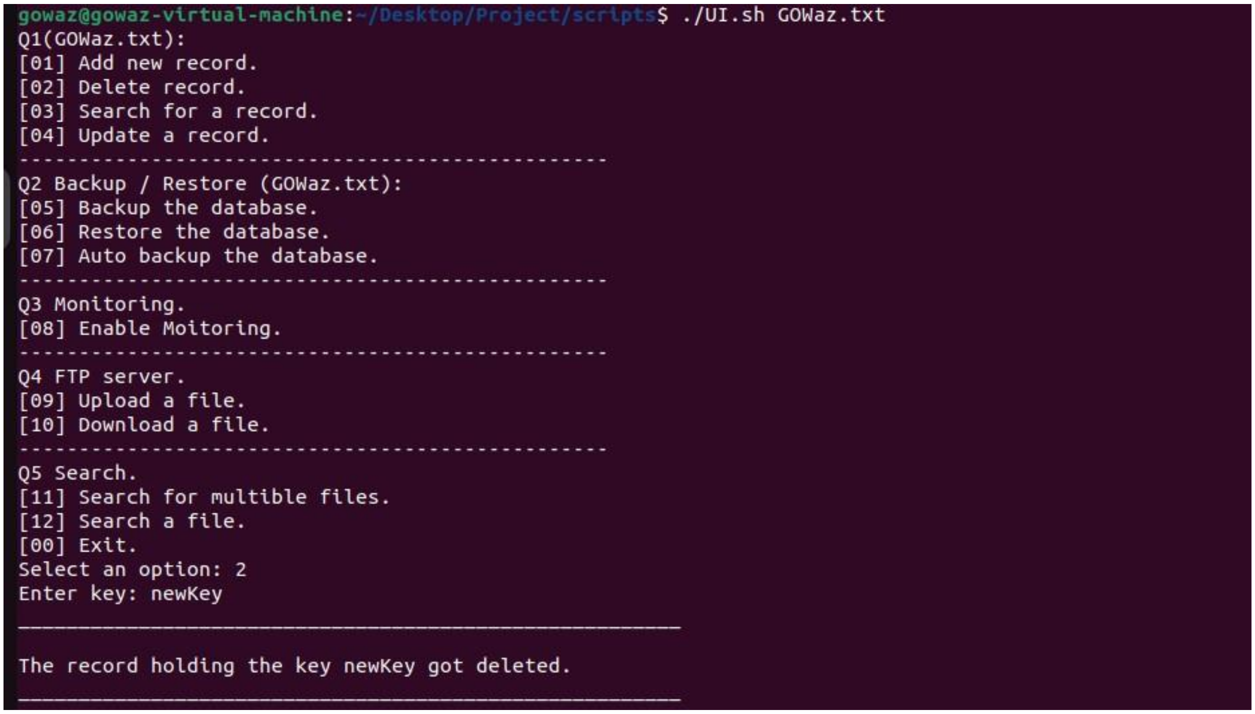
```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 1
Enter key: newKey
Q1Functionalty.sh: line 12: [: newKey: unary operator expected
Enter value: new value
_________________________________________________________________

Adding new key to the database (/home/gowaz/Desktop/Project/database/GOWaz.txt)...... Done
_________________________________________________________________
```



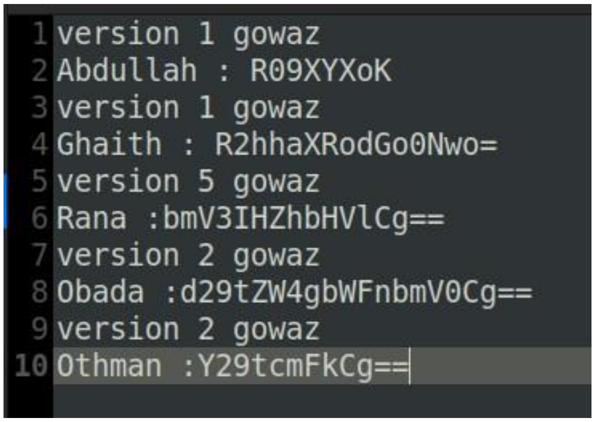
```
version 1 gowaz
newKey : bmV3IHZhbHVlCg==
```

ii. Delete record:

First, the function asks the user to enter the key, the function then goes and searches the database if the key matches any key from there, if it matches the key, it brings up the number of the line that the key was placed in, then it sends a command to delete both lines of the record and its version, and if it doesn't exist it displays "doesn't exist.".



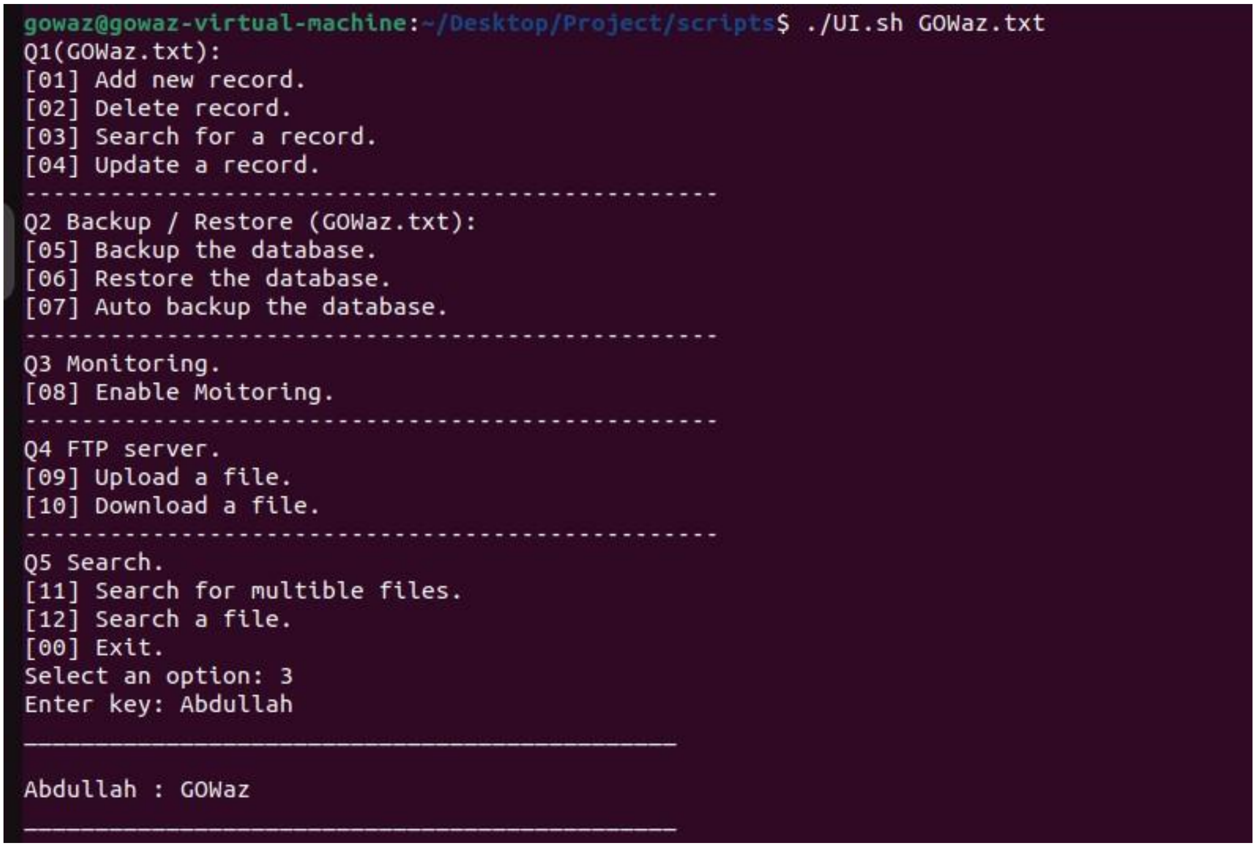
```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 2
Enter key: newKey
_______________________________________________________

The record holding the key newKey got deleted.
_______________________________________________________
```



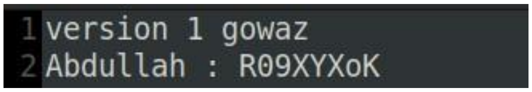
```
version 1 gowaz
Abdullah : R09XYXoK
version 1 gowaz
Ghaith : R2hhaXRodGo0Nwo=
version 5 gowaz
Rana :bmV3IHZhbHVlCg==
version 2 gowaz
Obada :d29tZW4gbWFnbmV0Cg==
version 2 gowaz
Othman :Y29tcmFkCg==
```

iii. Search record.

First, the function asks the user to enter the key, then it searches the database for any key that matches the entered key, if it matches a key, the function brings that record from the database, decodes the value, and displays the key with its decoded value for the user, and if it doesn't exist it displays "doesn't exist.".



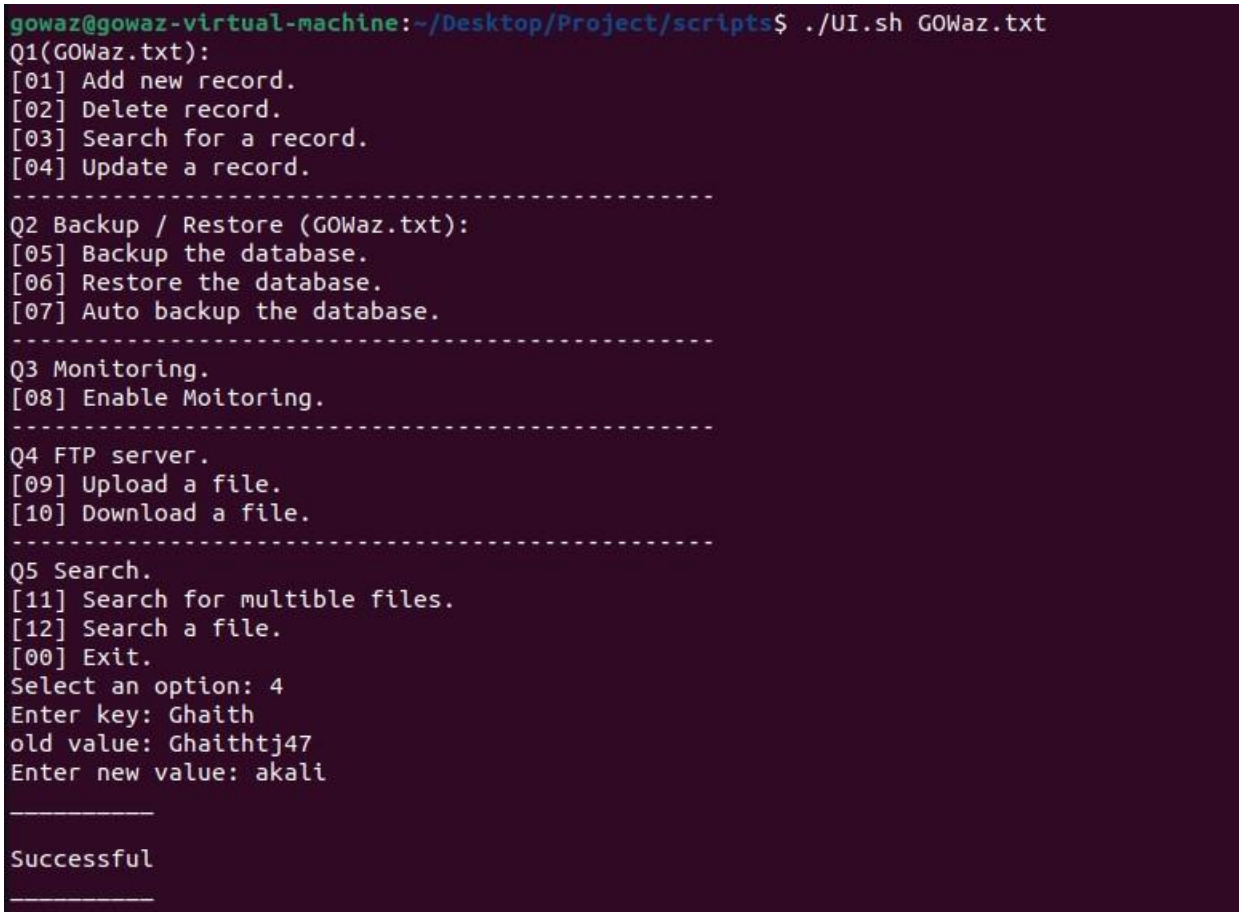
```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 3
Enter key: Abdullah
______________________________________________

Abdullah : GOWaz
______________________________________________
```



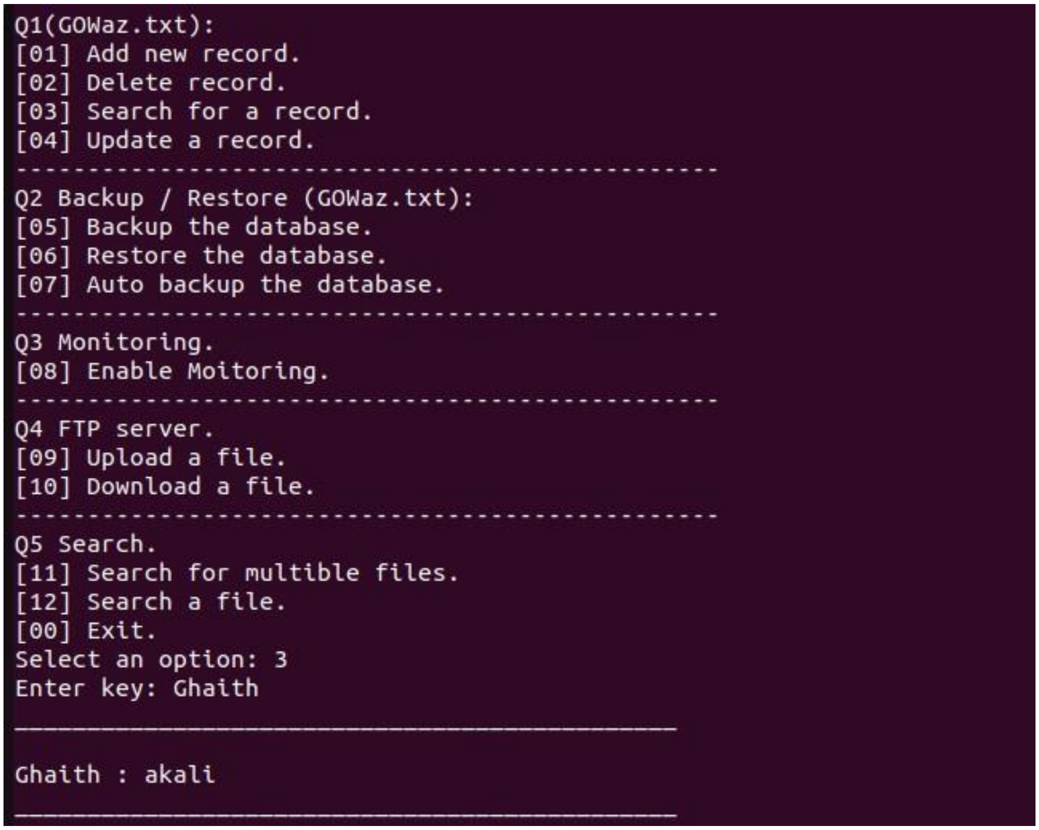
```
version 1 gowaz
Abdullah : R09XYXoK
```

iv. Edit record:

First, the function asks the user to enter the key, then it searches the database for the key that matches the entered key, and it brings the record with its value and what line in the database it's in, then it asks the user for the new value, after that it sends the new value and it updates the version number, of course, we send the number of the line we want to edit, so if there are to keys holding the same value, we don't change the unwanted key.

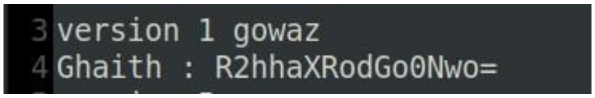


```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 4
Enter key: Ghaith
old value: Ghaithtj47
Enter new value: akali
__________

Successful
__________
```

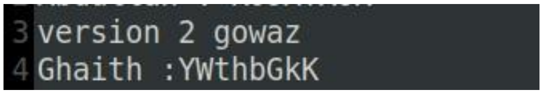


```
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 3
Enter key: Ghaith
______________________________________________

Ghaith : akali
______________________________________________
```

Before



```
version 1 gowaz
Ghaith : R2hhaXRodGo0Nwo=
```



```
version 2 gowaz
Ghaith :YWthbGkK
```

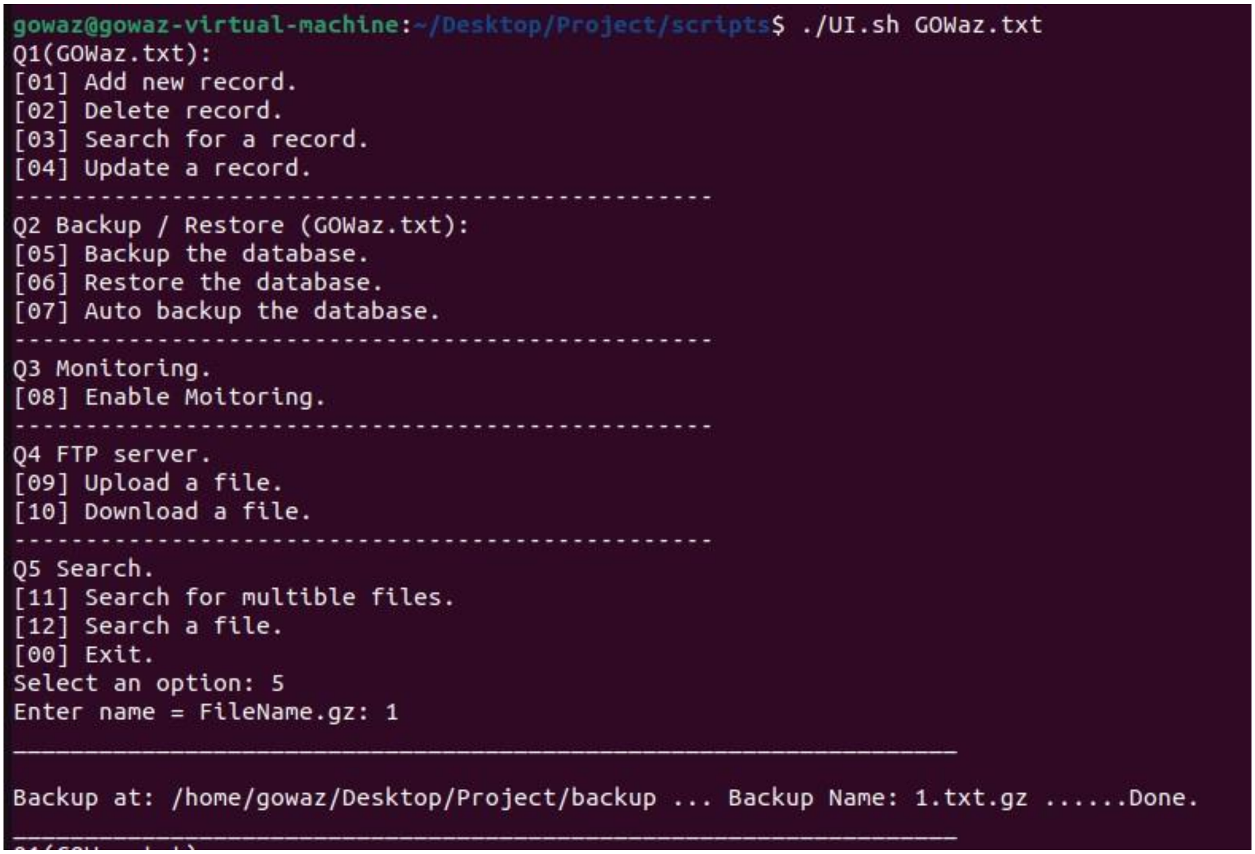
After

Q2:

We created one script it has three functions:

v. Backup database:

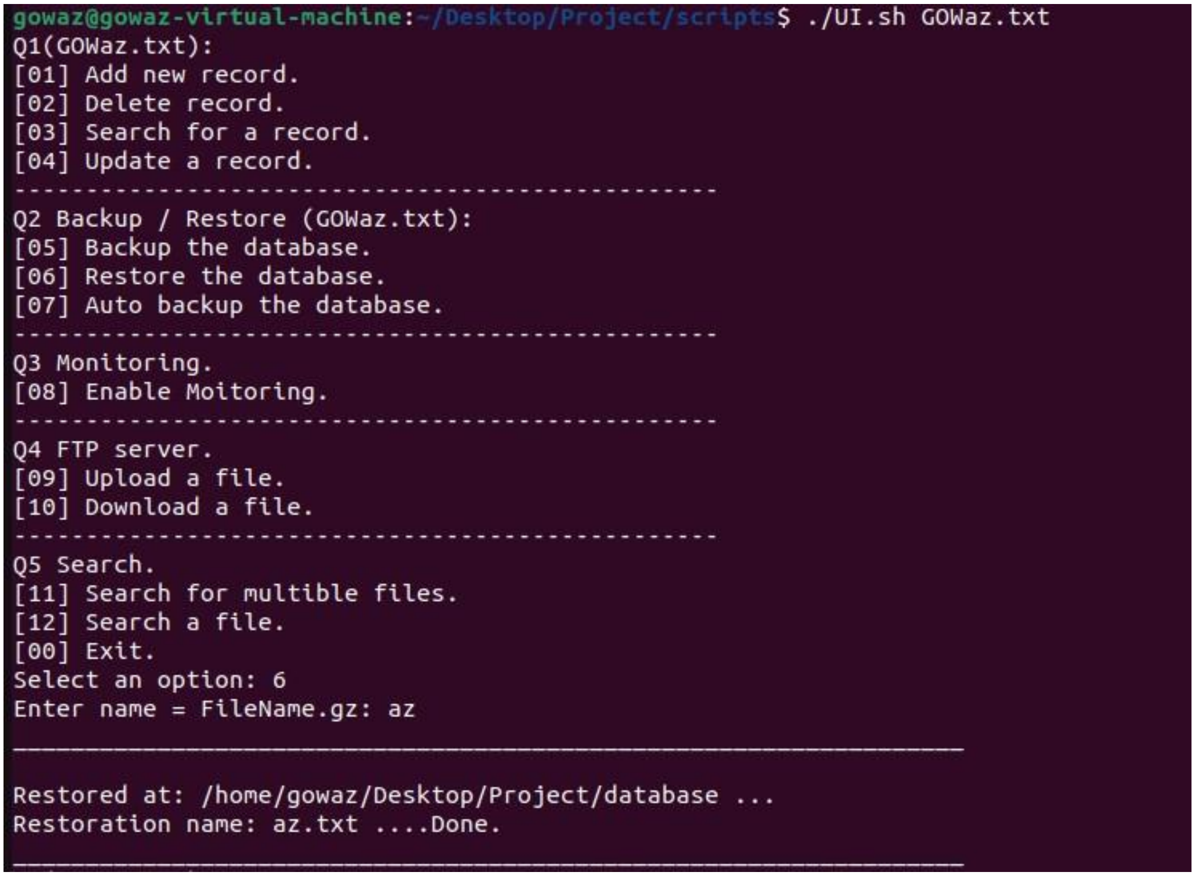
It asks the user to input the backup name, and it backs the database to the backup folder without deleting the original file.



```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 5
Enter name = FileName.gz: 1
__________________________________________________________________

Backup at: /home/gowaz/Desktop/Project/backup ... Backup Name: 1.txt.gz ......Done.
__________________________________________________________________
```

vi. Restore database:

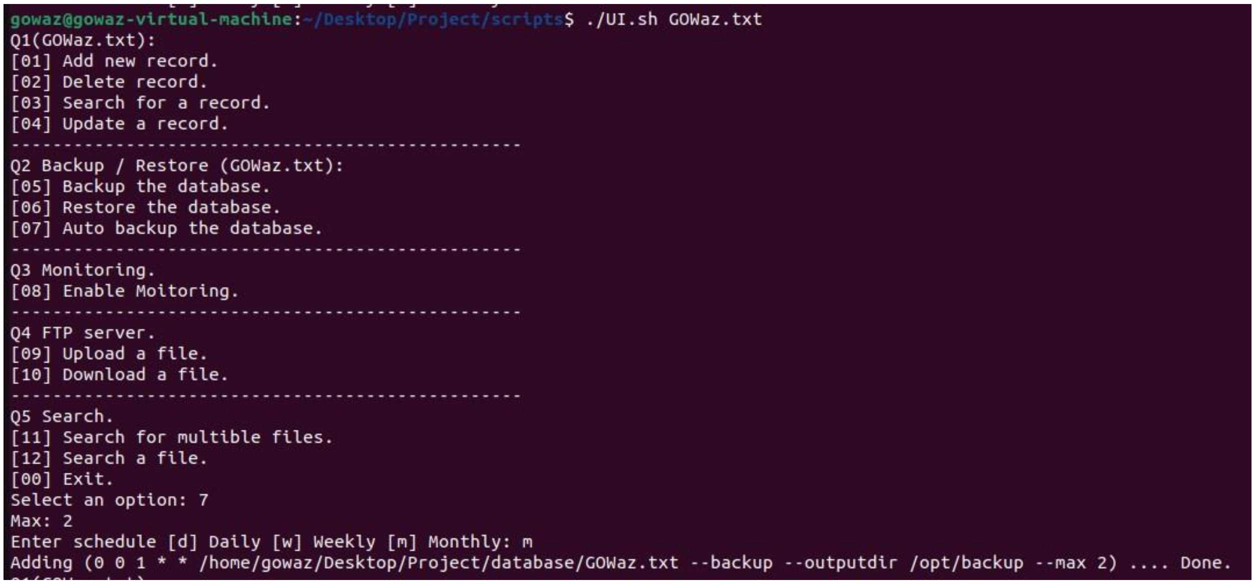
It asks the user to input the name of the backup name he wants to restore, then it restores it and puts it in the database folder.



```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 6
Enter name = FileName.gz: az
------------------------------------------------------------------

Restored at: /home/gowaz/Desktop/Project/database ...
Restoration name: az.txt ....Done.

------------------------------------------------------------------
```

vii. Auto backup for the database:

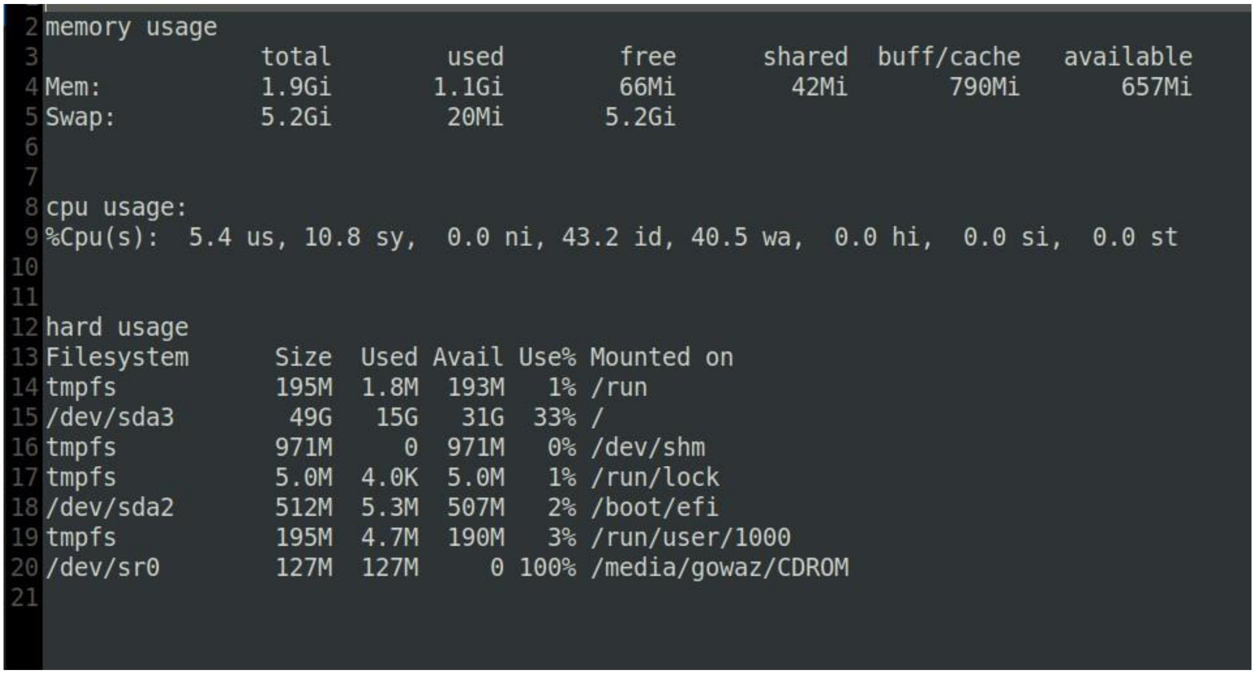
The function asks the user to input the maximum number of backups he wants, then it checks that it’s not equal to zero, then it asks the user to input the schedule Daily, Weekly, or Monthly, and depending on the choice it’ll run another script using crontab the other script it’ll checks the number of files in the backup folder if it’s bigger than max it’ll delete the oldest backups and create new backup holding the:
date of the machine_database name.txt.gz, and if it’s less than max then it creates a new backup same as before.



```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 7
Max: 2
Enter schedule [d] Daily [w] Weekly [m] Monthly: m
Adding (0 0 1 * * /home/gowaz/Desktop/Project/database/GOWaz.txt --backup --outputdir /opt/backup --max 2) .... Done.
```

Q3:

We created one script that contains four functions, each function calls a status command RAM, CPU, HARD, and a function that calls the other three, and create a file with the date that the script runs at and put the result in that file.



```
memory usage
               total        used        free      shared  buff/cache   available
Mem:           1.9Gi       1.1Gi        66Mi        42Mi       790Mi       657Mi
Swap:          5.2Gi        20Mi       5.2Gi


cpu usage:
%Cpu(s):  5.4 us, 10.8 sy,  0.0 ni, 43.2 id, 40.5 wa,  0.0 hi,  0.0 si,  0.0 st


hard usage
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           195M  1.8M  193M   1% /run
/dev/sda3        49G   15G   31G  33% /
tmpfs           971M     0  971M   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2       512M  5.3M  507M   2% /boot/efi
tmpfs           195M  4.7M  190M   3% /run/user/1000
/dev/sr0        127M  127M     0 100% /media/gowaz/CDROM
```

Q4:

1- To set up the FTP server we used the vsftpd server which is widely regarded as the fastest and most secure FTP server for UNIX-like systems.
First, we need a new machine to be the FTP server so we installed Ubuntu then we installed vsftpd and ufw, and FTP through this instruction.

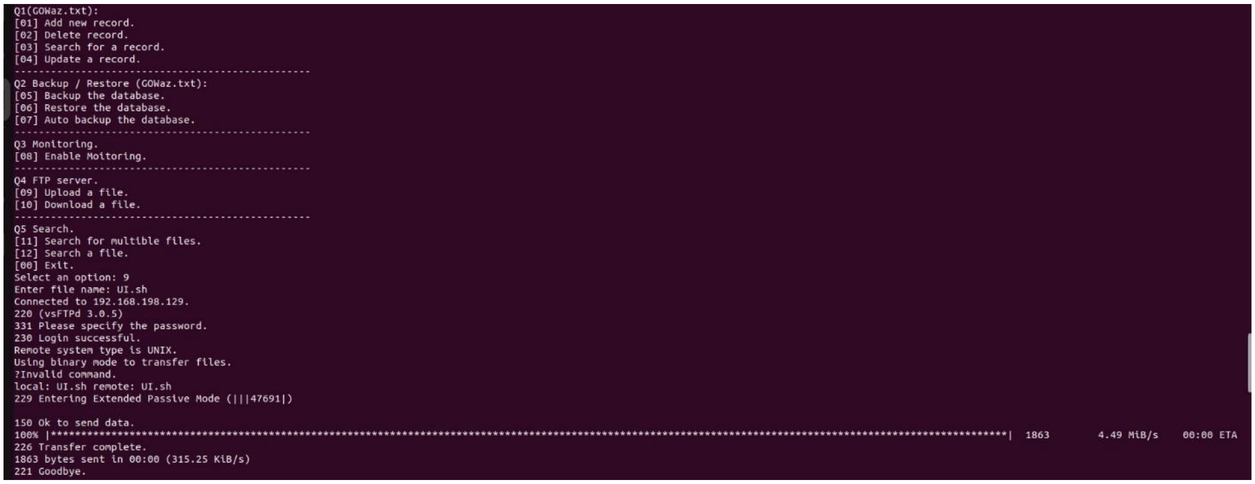
```
sudo apt install vsftpd ufw ftp -y
```

In our case the default configuration of vsftpd is fine, but we still need to make sure FTP traffic is allowed, so we executed these instructions:

```
sudo ufw allow 20/tcp
sudo ufw allow 21/tcp
sudo ufw allow 990/tcp
sudo ufw allow 40000:50000/tcp
```

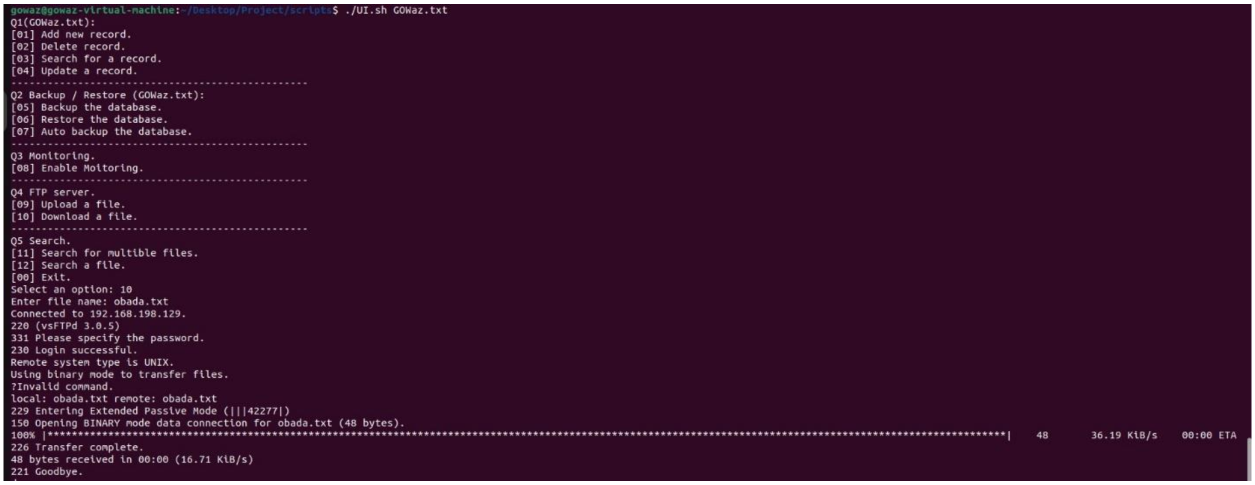
then we created a user who is going to use the FTP access.

2-In order to upload files from company pcs to our FTP server we need first to open a connection between them and use FTP user information (user name and password) to access the server,
then the employee has to enter his file's location and name, so we create the following script to do this thing:



```
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 9
Enter file name: UI.sh
Connected to 192.168.198.129.
220 (vsFTPd 3.0.5)
331 Please specify the password.
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
?Invalid command.
local: UI.sh remote: UI.sh
229 Entering Extended Passive Mode (|||47691|)

150 Ok to send data.
100% |****************************************************************************************************************************************************************|  1863        4.49 MiB/s    00:00 ETA
226 Transfer complete.
1863 bytes sent in 00:00 (315.25 KiB/s)
221 Goodbye.
```

3-In order to download files from the FTP server we have to do the same previous connection process but here the employee should enter the location and names of the files he wants, so we create the following script to do these things:

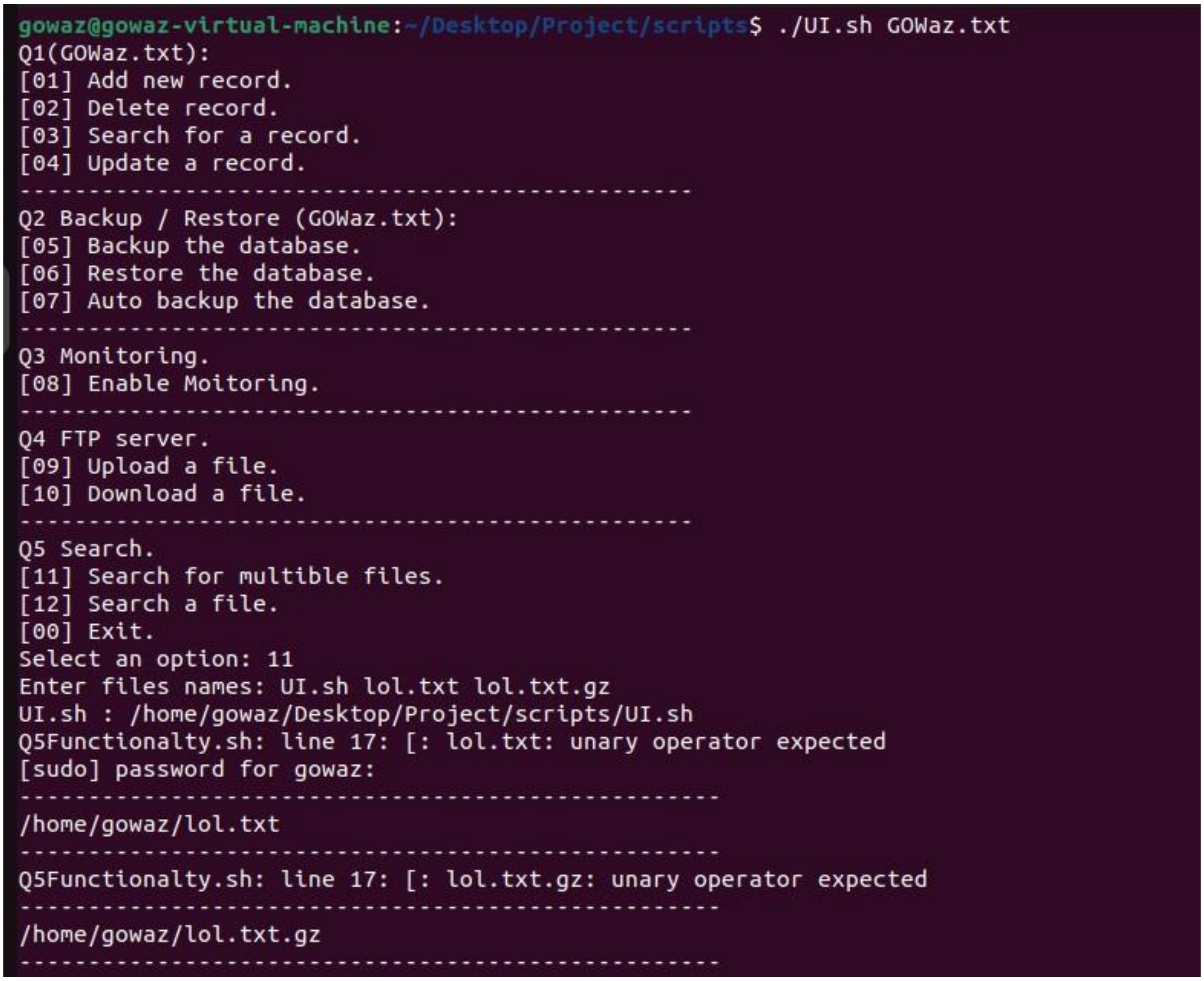


```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 10
Enter file name: obada.txt
Connected to 192.168.198.129.
220 (vsFTPd 3.0.5)
331 Please specify the password.
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
?Invalid command.
local: obada.txt remote: obada.txt
229 Entering Extended Passive Mode (|||42277|)
150 Opening BINARY mode data connection for obada.txt (48 bytes).
100% |****************************************************************************************************************************************************************|    48       36.19 KiB/s    00:00 ETA
226 Transfer complete.
48 bytes received in 00:00 (16.71 KiB/s)
221 Goodbye.
```

Q5:

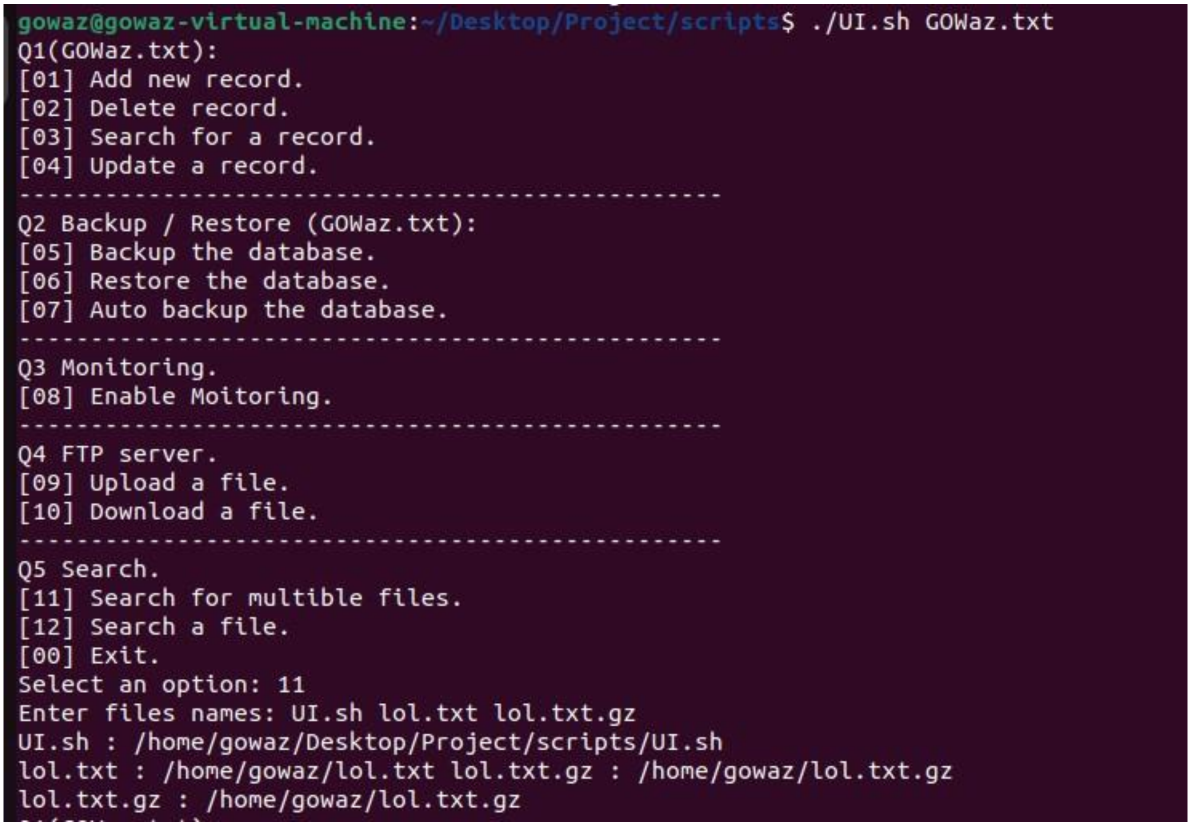
We created one script with two functions, one for searching multiple files and displaying the path of the searched files, and also saves the search result so if the user searched for the same file again the search will be faster it displays the result right away, and one to search for file info and displays the first three columns.

Here the script takes time to search for these files:

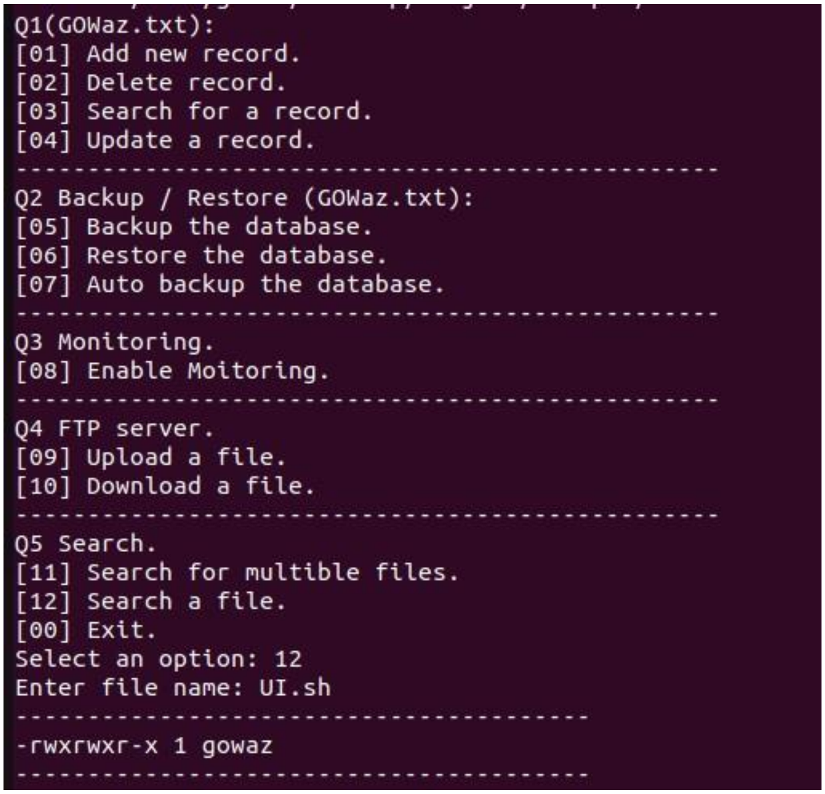


```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 11
Enter files names: UI.sh lol.txt lol.txt.gz
UI.sh : /home/gowaz/Desktop/Project/scripts/UI.sh
Q5Functionalty.sh: line 17: [: lol.txt: unary operator expected
[sudo] password for gowaz:
---------------------------------------------------
/home/gowaz/lol.txt
---------------------------------------------------
Q5Functionalty.sh: line 17: [: lol.txt.gz: unary operator expected
---------------------------------------------------
/home/gowaz/lol.txt.gz
---------------------------------------------------
```

Here the script instantly displays the result because of the previous search.



```
gowaz@gowaz-virtual-machine:~/Desktop/Project/scripts$ ./UI.sh GOWaz.txt
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 11
Enter files names: UI.sh lol.txt lol.txt.gz
UI.sh : /home/gowaz/Desktop/Project/scripts/UI.sh
lol.txt : /home/gowaz/lol.txt lol.txt.gz : /home/gowaz/lol.txt.gz
lol.txt.gz : /home/gowaz/lol.txt.gz
```



```
Q1(GOWaz.txt):
[01] Add new record.
[02] Delete record.
[03] Search for a record.
[04] Update a record.
-------------------------------------------------
Q2 Backup / Restore (GOWaz.txt):
[05] Backup the database.
[06] Restore the database.
[07] Auto backup the database.
-------------------------------------------------
Q3 Monitoring.
[08] Enable Moitoring.
-------------------------------------------------
Q4 FTP server.
[09] Upload a file.
[10] Download a file.
-------------------------------------------------
Q5 Search.
[11] Search for multible files.
[12] Search a file.
[00] Exit.
Select an option: 12
Enter file name: UI.sh
----------------------------------------
-rwxrwxr-x 1 gowaz
----------------------------------------
```

UI.sh

Tuesday, December 20, 2022 11:20 PM

```
#!/bin/bash
touch /home/gowaz/Desktop/Project/database/$1
file="/home/gowaz/Desktop/Project/database/$1"
. Q1Functionalty.sh
. Q2Functionalty.sh
. Q3Functionalty.sh
. Q4Functionalty.sh
. Q5Functionalty.sh
function auto_monitoring () {
  #crontab -u $USER -l > temp
  crontab -l > temp
  echo "00 12 * * * ./Q3Functionalty.sh" >> temp

  #crontab -u $USER temp
  crontab temp

  rm temp
  echo "__________________________________________________________________"
  echo " "
  echo "Auto Monitoring is enabled."
  echo "__________________________________________________________________"
}
option=20
while [ $option -ne 0 ]
do
echo "Q1($1):"
echo "[01] Add new record."
echo "[02] Delete record."
echo "[03] Search for a record."
echo "[04] Update a record."
echo "-------------------------------------------------"
echo "Q2 Backup / Restore ($1):"
echo "[05] Backup the database."
echo "[06] Restore the database."
echo "[07] Auto backup the database."
echo "-------------------------------------------------"
echo "Q3 Monitoring."
echo "[08] Enable Moitoring."
echo "-------------------------------------------------"
echo "Q4 FTP server."
echo "[09] Upload a file."
echo "[10] Download a file."
echo "-------------------------------------------------"
echo "Q5 Search."
echo "[11] Search for multible files."
echo "[12] Search a file."
echo "[00] Exit."
echo -n "Select an option: "
read option
case $option in
  1)
    insert_new_record "$file"
    ;;
  2)
    delete_record "$file"
    ;;
  3)
    search_record "$file"
    ;;
  4)
    edit_record "$file"
    ;;
  5)
    backup_database "$file"
    ;;
  6)
    restore_database
    ;;
  7)
    auto_backup "$file"
    ;;
  8)
    auto_monitoring
    ;;
  9)
    putt
    ;;
  10)
    gett
    ;;
  11)
    search_multible_files
    ;;
  12)
    search_file
    ;;
  *)
```

```
    echo "Exit"
    ;;
esac
done
```

Q1Funcionalty.sh

Tuesday, December 20, 2022 11:20 PM

```
#!/bin/bash
function insert_new_record () {
    echo -n "Enter key: "
    read key
    info=$(grep -w "$key" $1)
    key2=`echo $info| cut -d ":" -f 1`
    if [ $key == $key2 ]
    then
        echo "Already exist"
    else
        echo -n "Enter value: "
        read value

        coded=`echo $value|base64`

        echo "version 1 $USER" >> $1
        echo "$key : $coded" >> $1

        echo
"__________________________________________________________________"
        echo " "
        echo "Adding new key to the database ($1)...... Done"
        echo
"__________________________________________________________________"
        echo " "
    fi
}
function delete_record () {
    echo -n "Enter key: "
    read key
    info=$(grep -n "$key" $1)

    key2=`echo $info| cut -d ":" -f 2`

    line=`echo $info| cut -d ":" -f 1`

    line2=$(expr $line - 1)
    if [ $key == $key2 ]
    then
        sed -i.bak -e "$line d;$line2 d" $1

        echo "_______________________________________________________"
        echo " "
        echo "The record holding the key $key got deleted."
        echo "_______________________________________________________"
        echo " "
    else
        echo "__________"
        echo " "
        echo "Does not exist"
        echo "__________"
        echo " "
    fi
}
# grep -n
# grep -B 1
function search_record () {

    echo -n "Enter key: "
    read key

    value=$(grep -w "$key" $1)

    key2=`echo $value| cut -d ":" -f 1`

    if [ $key == $key2 ]
    then
        decoded=`echo $value| cut -d ':' -f2-`

        echo "______________________________________________"
        echo " "
        echo "$key : $(echo $decoded|base64 -d)"
        echo "______________________________________________"
        echo " "
    else
        echo "__________"
        echo " "
        echo "Does not exist"
        echo "__________"
        echo " "
    fi
```

```
}
function edit_record () {

    echo -n "Enter key: "
    read key

    info=$(grep -n "$key" $1)

    key2=`echo $info| cut -d ":" -f 2`

    line=`echo $info| cut -d ":" -f 1`

    line2=$(expr $line - 1)

    fullInfo=$(grep -B 1 "$key" $1)

    number=`echo $fullInfo| cut -d " " -f 2`

    newNumber=$(expr $number + 1)


    if [ $key == $key2 ]
    then

        coded=`echo $info| cut -d ':' -f3-`

        echo "old value: $(echo $coded|base64 -d)"

        echo -n "Enter new value: "
        read newValue

        newCoded=`echo $newValue|base64`

        sed -i "$line2 s/$number/$newNumber/" $1
        sed -i "$line s/$coded/$newCoded/" $1

        echo "__________"
        echo " "
        echo "Successful"
        echo "__________"
        echo " "
    else
        echo "__________"
        echo " "
        echo "Does not exist"
        echo "__________"
        echo " "
    fi
}
```

Q2AutoBackup.sh

Tuesday, December 20, 2022 11:20 PM

```
#!/bin/bash
path="/opt/backup"
number=$(ls $path | wc -l)
if [ $number >= $2 ]
then
    newPath=`echo $(ls $path | head -n 1)`
    rm $path/$newPath
    name=`echo $1|cut -d "/" -f 7`
    fullName=$(echo "$(date)_$name.gz")
    gzip -c $1 > $path/$fullName
elif [ $number < $2 ]
then
    name=`echo $1|cut -d "/" -f 7`
    fullName=$(echo "$(date)_$name.gz")
    gzip -c $1 > $path/$fullName
fi
```

Q2Functionalty.sh

Tuesday, December 20, 2022 11:20 PM

```
#!/bin/bash
path="/home/gowaz/Desktop/Project/backup"
path2="/home/gowaz/Desktop/Project/database"
function backup_database () {
    echo -n "Enter name: "
    read name

    fullName=`echo "$name.txt.gz"`

    gzip -c $1 > $path/$fullName
    echo "__________________________________________________________________"
    echo " "
    #echo "Backup at: $(readlink -f $fullName)"
    echo -n "Backup at: $path ... "
    echo "Backup Name: $fullName ......Done."
    echo "__________________________________________________________________"
}
function restore_database () {
    echo -n "Enter name: "
    read name

    fullName=`echo "$name.txt.gz"`
    newName=`echo "$name.txt"`

    gzip -d $path/$fullName > $path2/$newName
    rm $path/$newName
    echo "__________________________________________________________________"
    echo " "
    #echo "Restore at: $(readlink -f $newName)"
    echo "Restored at: $path2 ... "
    echo "Restoration name: $newName ....Done."
    echo "__________________________________________________________________"
}
function auto_backup () {

    read -p "Max: " max

    timee="* * * * *"

    if [ $max -eq 0 ]
    then
        ehco "Max can't be 0"
    else

    read -p "Enter schedule [d] Daily [w] Weekly [m] Monthly: " choice

    case $choice in
```

```
        d)
          timee="0 0 * * *"
          ;;
        w)
          timee="0 0 * * 0"
          ;;
        m)
              timee="0 0 1 * *"
          ;;
        *)
          echo "Wrong input!"
          ;;
    esac
    #crontab -u $USER -l > temp
    crontab -l > temp

    echo "$timee ./Q2AutoBackup.sh $1 $max" >> temp

    #crontab -u $USER temp
    crontab temp

    rm temp

    echo "Adding ($timee $1 --backup --outputdir /opt/backup --max $max) .... Done."

    fi
}
```

Q3Functionalty.sh

Tuesday, December 20, 2022   11:20 PM

```
#!/bin/bash

function memory () {
    echo ""
    echo "memory usage"
    free -h
    echo ""
}

function cpuusage(){
    echo ""
    echo "cpu usage:"
    top -b -n 1 -d1 | grep "Cpu(s)"
    echo ""
}


function hard() {
    echo ""
    echo "hard usage"
    df -h
    echo ""
}

function e() {
    memory
    cpuusage
    hard
}
#function start () {
    #while [ 0 ]
    #do
        #while [ $(date +%T) = 23:52:00 ]
        #do
                #touch  /home/gowaz/Desktop/q.$(date +%
T).txt
                #e > /home/gowaz/Desktop/q.$(date +%
T).txt
        #done
    #done
#}
touch  /home/gowaz/Desktop/q.$(date +%T).txt
e > /home/gowaz/Desktop/q.$(date +%T).txt
```

Q4Functionalty.sh

Tuesday, December 20, 2022 11:20 PM

```
#!/bin/bash
host="192.168.198.129"
user="gow"
password="2048"
function putt () {
    echo -n "Enter file name: "
    read file
    path='.'
    ftp -inv $host <<-EOF
    user $user $password
    put $file
    bye
    EOF
}
#cd $path
function gett () {
    echo -n "Enter file name: "
    read file
    path='.'
    ftp -inv $host <<-EOF
    user $user $password
    get $file
    bye
    EOF
}
```

Q5Functionalty.sh

Tuesday, December 20, 2022 11:20 PM

```bash
#!/bin/bash
touch searshHistory.txt
sh="searshHistory.txt"
function search_multible_files () {
    echo -n "Enter files names: "
    declare -a fileNames
    read fileNames
    for i in ${fileNames[@]}
    do
    info=$(grep -w "$i" $sh)
    key=`echo $info| cut -d ":" -f 1`

    if [ $i == $key ]
    then
        echo $info
    else
        found=$(sudo find / -type f -name "$i" 2>/dev/null & wait)
        echo "---------------------------------------------------"
        echo "$found"
        echo "---------------------------------------------------"
        echo "$i : $found" >> $sh
    fi
    done
}
function search_file () {
    echo -n "Enter file name: "
    read fileName
    info=$(find / -name "$fileName" -exec ls -la {} \; 2>/dev/null)
    echo "----------------------------------------"
    echo $info|cut -d " " -f 1,2,3
    echo "----------------------------------------"
}
```