#!/bin/bash

# CLT.sh

# ~/Command Line Test/CLT.sh

#--------------------------------Welcome--------------------------------#

function welcome

{

clear

echo "------------------------------------------------"

echo "| Welcome To Command Line Test By Aditya |"

echo "------------------------------------------------"

echo "Login/Register"

echo "1. Sign in"

echo "2. Sign up"

echo "3. Exit"

echo "Note: Script Exit on Timeout."

read -p "Please choose your option: " opt

if [[ $opt -eq 1 ]]

then

authentication

else

if [[ $opt -eq 2 ]]

then

sign\_up

else

if [[ $opt -eq 3 ]]

then

exit 0

else

echo "Invalid Option. Choose a correct option..."

welcome

fi

fi

fi

}

#--------------------------------Creating user.csv && password.csv--------------------------------#

if [ ! -f "user.csv" ] # check if user.csv file exists

then

touch user.csv

fi

if [ ! -f "password.csv" ] # check if password.csv file exists

then

touch password.csv

fi

#--------------------------------Authentication--------------------------------#

function authentication

{

clear

echo "------------------------------------------------"

echo "| Sign in |"

echo "------------------------------------------------"

# read -p "Enter Your Username: " username

# read -sp "Enter Your Password: " password

# if [[ $username -eq "Aditya" ]] && [[ $password -eq "1234" ]]

# then

# log\_in

# else

# echo -e "\nInvalid Credentials. Try Again After 10 SEC..."

# countdown

# clear

# authentication

# fi

read -p "Enter the username: " user # read the username

uname=(`cat user.csv`) # retrieve all username and store in array

uname\_len=${#uname[@]}

pass=(`cat password.csv`) # retrieve all password in encoded format and store in array

index=""

for i in `seq 0 $(($uname\_len-1))` # loop to find if entered username is present in user.csv file

do

if [ "$user" = "${uname[$i]}" ]

then

index=$i # save index value of the username matched in file

fi

done

if [ -n "$index" ] # check if index variable is not null meaning user is present

then

echo "Username Matched!!!"

read -sp "Enter your password: " pass1 # read password from user

else

echo "You are not signed up."

echo "Please Sign up."

iloading

sign\_up

fi

if [ $pass1 = `echo ${pass[$index]} | base64 --decode` ] # check if correct password is entered after decoding it from password.csv file

then

check=1

echo -e "\nPassword Matched."

echo "Signed in successfully!!!"

sleep 2

sign\_in

else

echo -e "\nPlease enter the correct Password.Try again..."

iloading

sleep 1

authentication

fi

}

#--------------------------------Sign in--------------------------------#

function sign\_in

{

clear

echo "------------------------------------------------"

echo " Welcome to your Test $user "

echo "------------------------------------------------"

echo "1. Take Test"

echo "2. View Test"

echo "3. Back"

read -p "Please choose your option: " op

if [[ $op -eq 1 ]]

then

take\_test

else

if [[ $op -eq 2 ]]

then

view\_test

else

if [[ $op -eq 3 ]]

then

welcome

else

echo "Invalid Option. Try Again..."

sleep 1

sign\_in

fi

fi

fi

}

#--------------------------------Take Test--------------------------------#

function take\_test

{

qbank\_lines=`cat questionbank.txt | wc -l` # store number of lines in question bank file

for i in `seq 5 5 $qbank\_lines` # loop to iterate through the question set each of five lines

do

clear

cat questionbank.txt | head -$i | tail -5 # display the question

for j in `seq 10 -1 1` # loop to iterate 10 times in reverse

do

echo -e "\r Enter the choice :\e[31m$j \e[0m \c" # intializing 10 second counter for user to provide the option

read -t 1 option

if [ -z "$option" ] # check if no option is selection means timeout

then

option="e"

else

break # else break the inner loop:

fi

done

echo $option >> user\_answer.txt # store option in a temporary file

echo "-------------------------"

done

clear

user\_ans=(`cat user\_answer.txt`) # store user selected option in an array

crrt\_ans=(`cat correctanswer.txt`) # store all the correct answers in an array

uans\_len=${#user\_ans[@]}

count=0

for i in `seq 0 $(($uans\_len-1))` # loop to check and compare the correct answers ans store in result.txt file

do

if [ ${user\_ans[$i]} = ${crrt\_ans[$i]} ]

then

echo "correct" >> result.txt

count=$(($count+1))

elif [ ${user\_ans[$i]} = "e" ]

then

echo "timeout" >> result.txt

else

echo "wrong" >> result.txt

fi

done

}

#--------------------------------View Test--------------------------------#

function view\_test

{

echo "This is also working"

}

#--------------------------------Sign Up--------------------------------#

function sign\_up

{

# clear

# echo "-----Sign Up and Register-----"

# read -p "Enter Your Name: " name

# read -p "Enter Your Username: " user

# read -p "Enter Your Password: " pass

# read -sp "Confirm Your Password: " cpass

# if [[ $pass -eq $cpass ]]

# then

# echo -e "\nYou're successfully signed up. Please Wait..."

# sleep 1

# echo -e "\nWe are redirecting you to Sign in"

# sleep 1

# loading

# sign\_in

# else

# echo -e "\nPassword Mismatch. Try Again After 10 SEC"

# countdown

# sign\_up

# fi

clear

echo "------------------------------------------------"

echo "| Sign up |"

echo "------------------------------------------------"

read -p "Enter the username: " user # read username

unames=(`cat user.csv`) # store all usernames from user file to array

unames\_len=${#unames[@]}

flag=1

for i in `seq 0 $(($unames\_len-1))` # loop to validate if entered username already exists or not

do

if [ $user = ${unames[$i]} ]

then

flag=0

fi

done

if [ $flag -eq 1 ] # if username does not exist request password from user

then

read -sp "Enter your password: " pass1

read -sp "Confirm your password: " pass2

if [ $pass1 = $pass2 ] # check for password confirmation

then

echo $user >> user.csv # save user to user.csv file

echo $pass2 | base64 >> password.csv # save encoded password in password.csv file

echo -e "\nUser registration succesfull!!!"

sleep 1

iloading

authentication

else

echo -e "\nPassword mismatch! Please enter correct password" # check for password mismatch & return to start menu

fi

else

echo "The $user username is already present" # check if user is available & return to start menu

echo "Kindly Sign in to proceed."

sleep 1

iloading

authentication

fi

}

#--------------------------------Infinite Loading--------------------------------#

function iloading

{

while true;

do

# Frame #1

printf "\r< Loading."

sleep 0.5

# Frame #2

printf "\r> Loading.."

sleep 0.5

# Frame #3

printf "\r> Loading..."

sleep 0.5

break

done

}

#--------------------------------Loading--------------------------------#

function loading

{

clear

echo "loading"

sleep 0.5

clear

echo "loading."

sleep 0.5

clear

echo "loading.."

sleep 0.5

clear

echo "loading..."

sleep 0.5

clear

}

#--------------------------------Countdown--------------------------------#

function countdown

{

hour=00

min=00

sec=10

while [ $hour -ge 0 ]; do

while [ $min -ge 0 ]; do

while [ $sec -ge 0 ]; do

echo -ne "$hour:$min:$sec\033[0K\r"

let "sec=sec-1"

sleep 1

done

sec=59

let "min=min-1"

done

min=59

let "hour=hour-1"

done

}

#--------------------------------Main--------------------------------#

welcome