Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2023), B.Sc. in CSE (Day)

Designing and implementation of a script command line for MCQ Test (Auto Evaluated).

Course Title: Operating System Lab
Course Code: CSE 310
Section: 201 DB

<u>Students Details</u>

Name	ID
Shamim Ahmned	201902067
Mohammad Shuvo	201902068
SK. Nahid	201902073

Submission Date: 20/06/2023
Course Teacher's Name: Jarin Tasnim Tonvi

<u>Lab Project Status</u>

Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

In This project we will designing and implementa of a script command line for MCQ Test (Auto Evaluated).The MCQ Test Command Line Script is a software project designed to facilitate the creation, administration, and auto-evaluation of multiple-choice question (MCQ) tests. The script provides a command-line interface that allows users to easily manage and conduct MCQ tests in an efficient and automated manner. It automates the process of evaluating the answers provided by the test takers and generates instant results, saving time and effort for both administrators and participants.

1.2 Motivation

The motivation behind developing the MCQ Test Command Line Script (Auto-Evaluated) project lies in addressing several challenges and improving the overall experience of conducting and evaluating MCQ tests.

Firstly, traditional manual evaluation of MCQ tests can be time-consuming, especially when dealing with a large number of participants. By automating the evaluation process, the script significantly reduces the time and effort required to grade tests. It eliminates the need for manual grading, allowing administrators to focus on other essential tasks.

Secondly, human error in manual evaluation can lead to inconsistencies and inaccuracies in grading. With the auto-evaluation feature of the script, all answers are assessed using the same predefined criteria. This ensures consistency in evaluating responses and provides accurate results, eliminating subjective bias.

Chapter 2

Problem Analysis

2.1 Problem Statement

The problem statement revolves around the limitations of manual evaluation and administration of MCQ tests, including time-consuming evaluation, potential for errors in grading, delayed feedback for participants, complex test administration procedures, and limited data analysis capabilities. These challenges hinder the efficiency and effectiveness of MCQ testing processes and call for the development of an automated solution to streamline evaluation, provide instant feedback, simplify administration, and enable data analysis.The existing manual process of administering and evaluating MCQ tests is time-consuming, error-prone, lacks instant feedback, involves complex administration procedures, and offers limited data analysis capabilities. Addressing these problems requires the development of an automated MCQ Test Command Line Script that streamlines the evaluation process, provides instant feedback, simplifies test administration, and enables data analysis for improved test management and participant experience.

2.2 Project Requirements

- Ubuntu OS(Operating System.
- A terminal for run the program.
- Notepad/VS Code to edit the shall script

Chapter 3

Implementation

3.1 Design

Initially the user will be provided with a sign-in option where pre-defined users will be allowed to log in. Upon successful login this tool will display questions for the user from existing data-base. It will also handle error conditions like time-out. This tool will also store answers provided by users for future verification. Provide a prompt for the user to sign-up and sign-in

- Sign In
- Take Test
- Sign up
- Exit

3.2 PROCEDURE

Algorithm 1: A script command line for MCQ Test

```
1 Define the functions: sign_up, sign_in, start_test, results, and header.
2 Display a welcome message and available options (signup, signin, exit).
3 Prompt the user to choose an option.
4 If signup is chosen then Prompt for a username,passward and signup.
5 If signin is chosen then Prompt for a username and pass login.
6 If the user signed in successfully, offer options to take the test or exit.
7 After the test, calculate the score and provide feedback on each question.
8 Display the total score and exit the program.
9 END
```

3.3 Implementation

Listing 3.1: A script command line for MCQ Test

```
#!/bin/bash

sign_up ()

{
    #sleep 3s
    header  # header part linked with sign_up part
    echo -e "\e[92m<<<<<\e[0m Signup \e[92m>>>>>\e[0m"
    users=(`cat user_name.csv`) # checking the content of user_name.c
    echo "Enter Username:"
    read username
    #echo "$((${#users[@]}))"
    #echo "${users[@]}"
    for user in `seq 0 $((${#users[@]}-1))`
    do
        userid=${users[$user]}
#       echo "$userid"

    if [ $userid = $username ] # checking the new username with the e
    then
        echo -e "\e[31mUsername already exists! Plase use different u
        sign_up  # if username already existing then it will redirect
    fi
    done
    pw=1
    while [ $pw -ne 0 ]
    do
        echo "Enter Password: "
        read -s password
#       echo "length ${#password}"
        if [ ${#password} -lt 6 ] # checking the password length less
        then
            echo -e "\e[31mPlease use atleast 6 character of password
        else
            pw=0
        fi
    done
    pass=1
    attempts=4
    while [ $pass -ne 0 -a $attempts -ne 0 ]
    do
        echo "Confirm Password: "
        read -s conf_password
        if [ $conf_password = $password ] # checking the password wit
        then
```

```
                pass=0
                echo -e "\e[32mCongratulation! Your Username & Password (

                echo $username >> user_name.csv  # storing the username i
                echo $conf_password >> password.csv # storing the passwor
                welcome
            else
                attempts=$(($attempts-1)) # if password is not match with
                echo -e "\e[31mWrong Password! Remaining Attempts\e[0m =

                if [ $attempts -eq 0 ] # if attempts equal to zero then i
                then
                    echo -e "\e[31mSorry! You crossed your maximum limit.
                    echo "Signup Again!"
                    sign_up
                fi
            fi
        done

    }

    sign_in ()

    {
        #sleep 3s
        header
        echo -e "\e[32m<<<<<\e[0m Signin \e[32m>>>>>\e[0m"
        users=(`cat user_name.csv`)  # storing the contents of user_name.
        found=0
        attempts=4 # 4 attempts are providing to the user
        while [ $found -ne 1 -a $attempts -ne 0 ]
        do
            echo -e "\e[34mEnter Username:\e[0m "
            read login_user
            for user in `seq 0 $((${#users[@]}-1))`
            do
                #echo ${users[$user]}

                if [ ${users[$user]} = $login_user ] # checking the usern
                then
                    found=1
                    position=$user # collecting the index value of 'user'
            #echo "User: $position"
                fi
            done
            if [ $found -eq 1 ]
            then
                echo -e "\e[92m:) Great!\e[0m"
        else
            echo -e "\e[31mUsername does't exist\e[0m"
            attempts=$(($attempts-1)) # if condition is false the att
            if [ $attempts -gt 0 ]
            then
                echo "Please try again"
                echo "You have $attempts attempts remaining"
            else
                echo -e "\e[31mSorry! You crossed your maximum limit.
                echo "Plaease Signup again"
               welcome  # after 4 attempts it will redirect to the si
            fi
        fi
    done

    password=(`cat password.csv`)   # storing the contents of passwor
    attempts=4
    found=0
    while [ $found -ne 1 -a $attempts -ne 0 ]
    do
        echo -e "\e[34mEnter Password:\e[0m "
        read -s login_pass
        echo
        #echo "Position: $position"  # position value is the same inc
        #echo "Pass Position: ${password[$position]}"
        if [ ${password[$position]} = $login_pass ]  # checking the c
        then
         #   echo "Password Matched"
            echo -e "\e[92mLogin Successful! You can start your exam.
            start_test # if condition true then it will go the test p
            found=1
        else
            echo -e "\e[31mWrong Password! Plase try again.\e[0m"
            attempts=$(($attempts-1))  # if condition false then atte
            if [ $attempts -gt 0 ]
            then
                echo "You have $attempts attempts remaining"
            else
                echo -e "\e[31mSorry! No more attempts. Please try la
                welcome # if 4 attempts over the it will be redirect
            fi
        fi

    done
}

start_test ()
{
```

```
header
echo -e "1) \e[32mTake the Test\e[0m"
echo -e "2) \e[31mExit\e[0m"
echo
read -p "Enter your choice: " choice
line=`cat question_bank.txt | wc -l`  # checking the line number

case $choice in
    1)
        for i in `seq 5 5 $line` # starting i value is 5 & it wil
        do
            #clear
            #sleep 2s
            header
            echo
            head -$i question_bank.txt | tail -5  # display the 5
            echo
            for j in `seq 10 -1 1` # ti
            do
                echo -e "\r\e[31mEnter the currect answer\e[0m \e
                read -t 1 ans # '-t' enable the time by 1 second
                if [ ${#ans} -ne 0 ]
                then
                    break
                fi
            done
    #       echo "word count: ${#ans}"

    #       read -p "Choose the currect answer : " ans
            if [ ${#ans} -eq 1 ]
            then
            echo "$ans" >> user_answer.txt  # user answer are sto
            else
            echo "No_Answer" >> user_answer.txt # if user not giv
            fi
            echo

            #echo "Next Question"
        done


    ;;

    2)
        exit
    ;;

    *) echo -e "\e[31mPlease choose currect option.\e[0m"
```

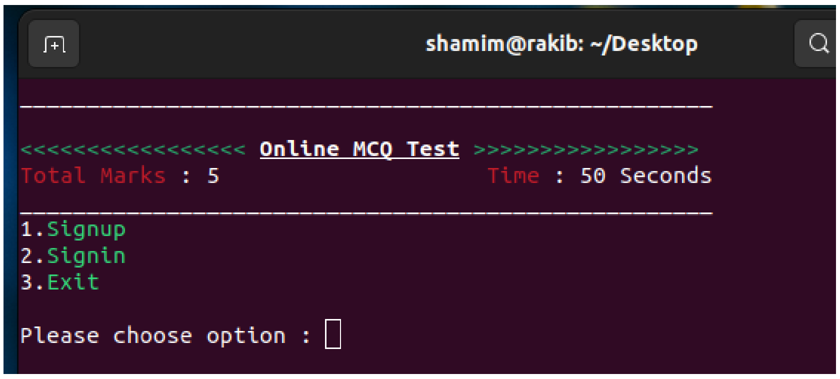
```
                start_test
    esac
    results
}

results ()

{
    header
    c_ans=(`cat currect_answer.txt | tr -s ' ' | cut -d ':' -f1`) #
    c_ans1=(`cat currect_answer.txt | tr -s ' ' | cut -d ':' -f2`) #
    u_ans=(`tail -5 user_answer.txt`) # 'u_ans' variable is storing t
    #echo "${c_ans[@]}"
    #echo "${u_ans[@]}"
    score=0

    for i in `seq 0 $((${#c_ans[@]}-1))`
    do
        if [ ${c_ans[i]} = ${u_ans[i]} ] # checking the user answer w
        then
            echo -e "Q$(($i+1)) Your answer is : \e[32m${c_ans[i]}
            echo -e "\e[32mQ$(($i+1))\e[0m \e[34mCurrect answer is
            echo
            score=$(($score+1)) # if condition true then score value
        else
            echo -e "Q$(($i+1)) Your answer is : \e[31m${u_ans[i]}
            echo -e "\e[32mQ$(($i+1))\e[0m \e[34mCurrect answer is
            echo
        fi
    done
        echo -e "\e[32mYour Total Score\e[0m: $score Marks"
        echo
        exit

}

header ()

{   sleep 2s  # it will wait for 2 seconds
    clear  # after 2 seconds will clear the display
        echo ____________________________________________________
        echo
        echo -e "\e[32m<<<<<<<<<<<<<<<<<\e[0m \e[1;4mOnline MCQ Test
        echo -e "\e[31mTotal Marks\e[0m : 5
\e[31mTime\e[0m : 50 Seconds"
        echo ____________________________________________________

}
```

```
#header

welcome ()
{
        header
        echo -e "1.\e[92mSignup\e[0m"
        echo -e "2.\e[92mSignin\e[0m"
        echo -e "3.\e[92mExit\e[0m"
        echo

        read -p "Please choose option : " choice

        case $choice in
                1) echo -e "\e[92mGreat! You are ready to Signup.\e[0
                    sign_up
                   ;;

                2) echo -e "\e[92mGreat! You are ready to Signin.\e[0
                    sign_in
                    ;;

                3) exit
                    ;;

                *) echo -e "\e[91mPlease choose currect option\e[0m"
                    welcome
                    ;;

        esac

}
```

Chapter 4

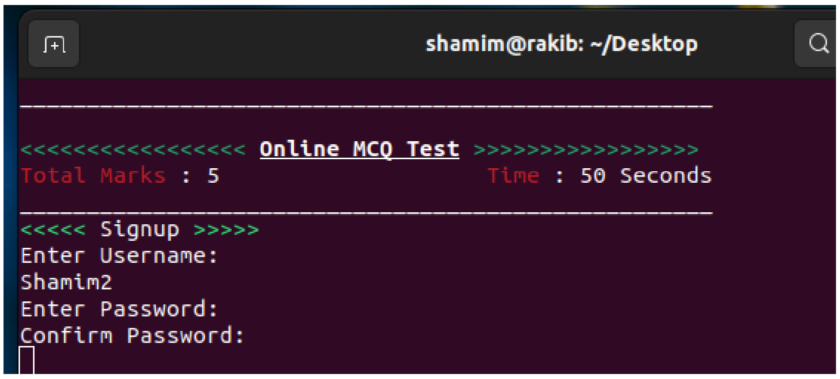
Result & Discussion

4.1 Result



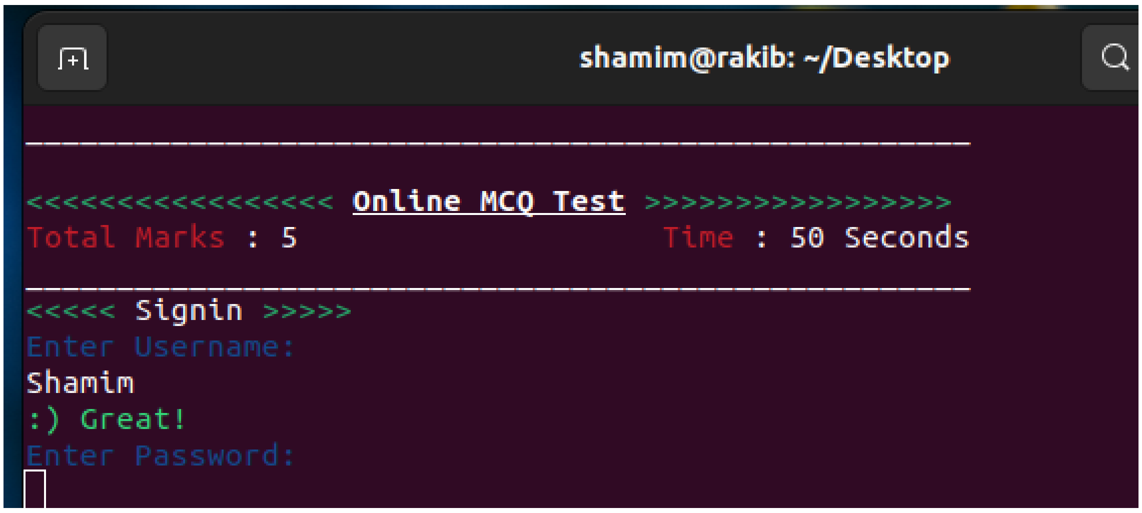
```
shamim@rakib: ~/Desktop
-----------------------------------------------------

<<<<<<<<<<<<<<<<< Online MCQ Test >>>>>>>>>>>>>>>>>
Total Marks : 5                    Time : 50 Seconds

-----------------------------------------------------
1.Signup
2.Signin
3.Exit

Please choose option :
```

Figure 4.1: Main Screen



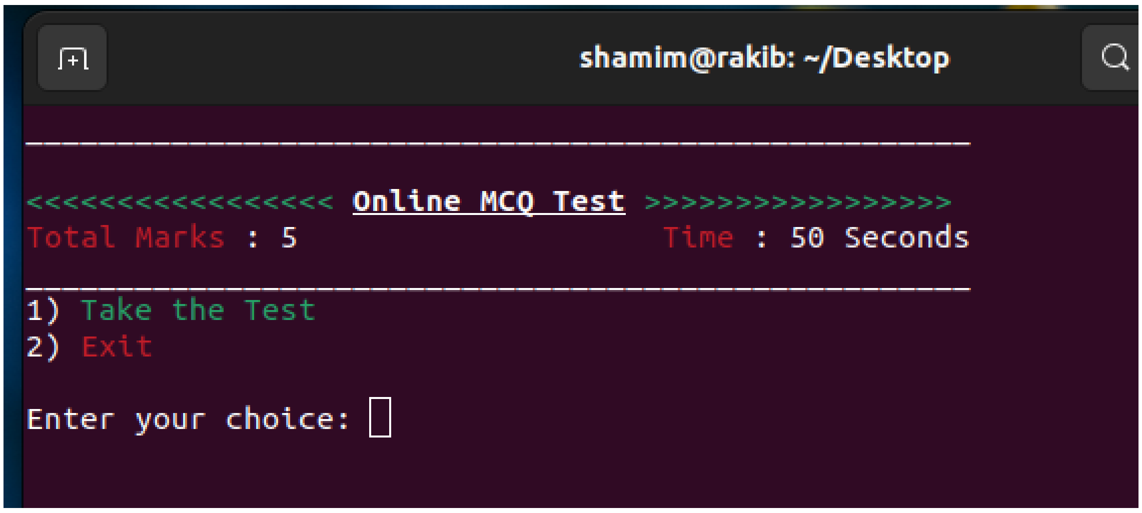
```
shamim@rakib: ~/Desktop
-----------------------------------------------------

<<<<<<<<<<<<<<<<< Online MCQ Test >>>>>>>>>>>>>>>>>
Total Marks : 5                    Time : 50 Seconds

-----------------------------------------------------
<<<<< Signup >>>>>
Enter Username:
Shamim2
Enter Password:
Confirm Password:
```

Figure 4.2: Sgin Up



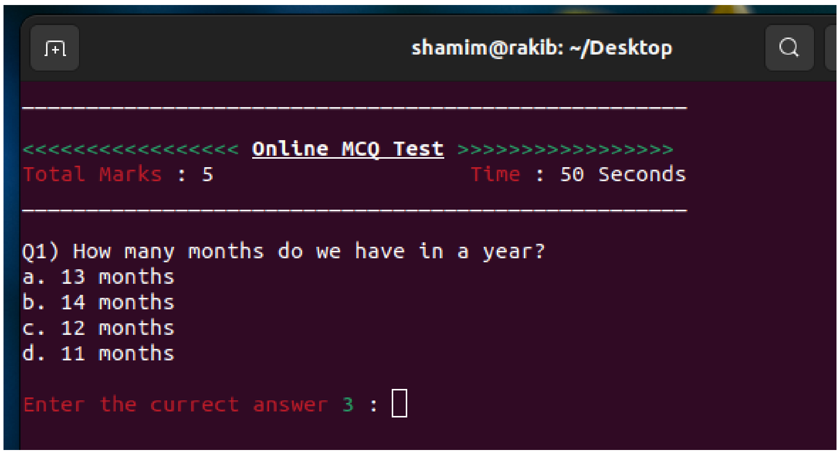
```
shamim@rakib: ~/Desktop
__________________________________________________________

<<<<<<<<<<<<<<<<<< Online MCQ Test >>>>>>>>>>>>>>>>>>
Total Marks : 5                            Time : 50 Seconds
__________________________________________________________
<<<<< Signin >>>>>
Enter Username:
Shamim
:) Great!
Enter Password:
```

Figure 4.3: Sign In



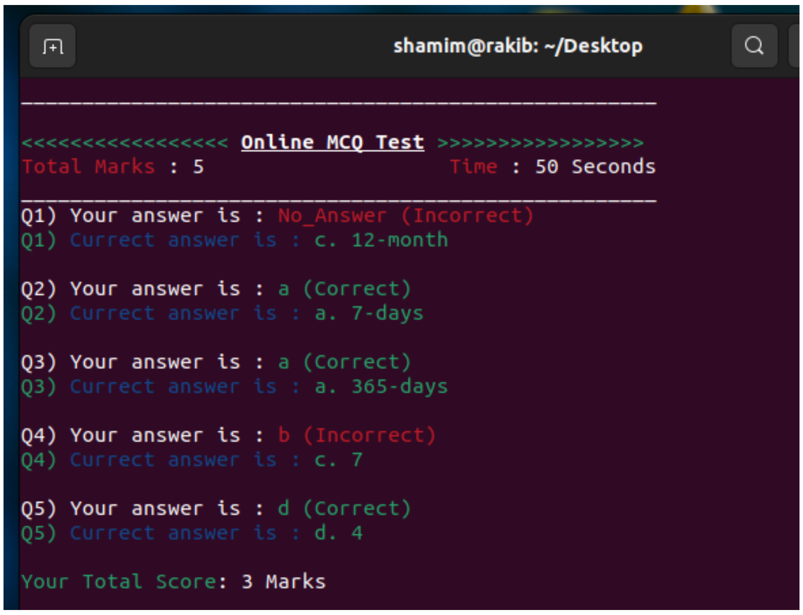
```
shamim@rakib: ~/Desktop
__________________________________________________________

<<<<<<<<<<<<<<<<<< Online MCQ Test >>>>>>>>>>>>>>>>>>
Total Marks : 5                            Time : 50 Seconds
__________________________________________________________
1) Take the Test
2) Exit

Enter your choice:
```

Figure 4.4: Start the test



```
shamim@rakib: ~/Desktop
----------------------------------------------------------

<<<<<<<<<<<<<<<<< Online MCQ Test >>>>>>>>>>>>>>>>>
Total Marks : 5                    Time : 50 Seconds

----------------------------------------------------------

Q1) How many months do we have in a year?
a. 13 months
b. 14 months
c. 12 months
d. 11 months

Enter the currect answer 3 :
```

Figure 4.5: Question and giving the answer



```
shamim@rakib: ~/Desktop
----------------------------------------------------------

<<<<<<<<<<<<<<<<< Online MCQ Test >>>>>>>>>>>>>>>>>
Total Marks : 5                    Time : 50 Seconds

----------------------------------------------------------
Q1) Your answer is : No_Answer (Incorrect)
Q1) Currect answer is : c. 12-month

Q2) Your answer is : a (Correct)
Q2) Currect answer is : a. 7-days

Q3) Your answer is : a (Correct)
Q3) Currect answer is : a. 365-days

Q4) Your answer is : b (Incorrect)
Q4) Currect answer is : c. 7

Q5) Your answer is : d (Correct)
Q5) Currect answer is : d. 4

Your Total Score: 3 Marks
```

Figure 4.6: Overall mark

4.2 Conclusion

The MCQ Test command-line [1] script is designed to provide a user-friendly interface for conducting multiple-choice question tests. The script allows users to sign up with a unique username and password or sign in if they already have an account. Once signed in, users can choose to take the test, which presents a series of questions and allows them to input their answers. After completing the test, the script evaluates the answers, calculates the score, and provides feedback on each question.

The script is designed to ensure data integrity by checking for existing usernames during signup and validating [2] & [3] passwords. It also includes error handling and prompts users with appropriate messages in case of invalid inputs or exceeded attempts. The implementation includes the use of external files to store user information, questions, correct answers, and user responses.

4.3 Scope of Future Work

will also support other features like predefined time per question, output reports etc. The idea of this project is to simulate such an online test interface using Linux Shell Scripting and commands. By implementing this Linux Shell Scripting Projects for Beginners Project will make you apply Shell programming constructs...

References

[1] Author initial. author surname, title. city: Publisher, year published, p. pages used.

[2] A. Rezi and M. Allam, ”Techniques in array processing by means of transformations, ” in Control and Dynamic Systems, Vol. 69, Multidemsional Systems, C. T. Leondes, Ed. San Diego: Academic Press, 1995, pp. 133-180.

[3] O. B. R. Strimpel, ”Computer graphics,” in McGraw-Hill Encyclopedia of Science and Technology, 8th ed., Vol. 4. New York: McGraw-Hill, 1997, pp. 279-283.