



Green University of Bangladesh

*Department of Computer Science and Engineering (CSE)
Semester: (spring, Year: 2024), B.Sc. in CSE (Day)*

Test Exam Taker

*Course Title: Operating System Lab
Course Code: CSE-310
Section: 213-D6*

Students Details

Name	ID
Nadib Rana	213002247
Shafin Mahud	213002084

*Submission Date: 15-06-2024
Course Teacher's Name: Abdullah Al Farhad*

<u>Lab Project Status</u>	
Marks:	Signature:
Comments:	Date:

Contents

1	Introduction	3
1.1	Overview.....	3
1.2	Motivation.....	3
1.3	Problem Definition.....	4
1.3.1	Problem Statement.....	4
1.3.2	Complex Engineering Problem.....	4
1.4	Design Goals/Objectives.....	5
1.5	Application.....	5
2	Design/Development/Implementation of the Project	6
2.1	Introduction.....	6
2.2	Project Details.....	6
2.2.1	Subsection_name.....	6
2.2.2	Feature Set.....	6
2.3	Implementation.....	7
2.3.1	Subsection_name.....	7
2.4	Algorithms.....	8
3	Performance Evaluation	10
3.1	Simulation Environment/ Simulation Procedure.....	10
3.1.1	Hardware and Software Setup.....	10
3.1.2	Simulation Procedure.....	11
3.2	Results Analysis/Testing.....	11
3.2.1	Execution Time.....	11
3.2.2	Accuracy.....	11
3.2.3	Result_portion_3.....	11
3.3	Results Overall Discussion.....	12

4	Conclusion	13
4.1	Discussion.....	13
4.2	Limitations.....	13
4.3	Scope of Future Work.....	14

Chapter 1

Introduction

1.1 Overview

This project of ours is mainly developed for students to take MCQ exam. Here we can set our own questions for the students and also see how many marks the student has scored in the exam. Here a student will be asked a maximum of 10 questions and 10 seconds will be allotted for each question. Once the 10 seconds pass, the student will not be able to write that answer.

In order to overcome these obstacles, our research presents a command-line interface application that automates the development, testing, and assessment of multiple-choice questions (MCQs). Our solution improves the user experience, assures accurate re- sponse evaluation, and speeds test administration by utilizing scripting.

1.2 Motivation

1. **Enhancing Communication:** Through the provision of an intuitive interface and optimized workflows, the application enhances interactivity between examiners and test takers.
2. **Education:** By automating the evaluation process, the application supports educational assessment and frees up instructors to concentrate more on mentorship and instruction.
3. **Professionalism:** Automated assessments provide fair and uniform grading, which raises the standard of the testing procedure.
4. **Time-Saving:** The application saves time for test administrators and takers alike by automating exam administration and evaluation.

1.3 Problem Definition

1.3.1 Problem Statement

The project intends to solve the demand for a user-friendly and effective command-line interface system for administering Multiple Choice Question (MCQ) tests. MCQ test manual evaluation can be laborious and error-prone. As a result, there is a need for an automated system that can guarantee accurate evaluation, expedite the test administration procedure, and give consumers immediate response.

This project's main objective is to provide a script-based application that makes it simple for users to design, administer, and assess multiple-choice questions (MCQs). The system seeks to increase MCQ test efficiency, accuracy, and scalability through test process automation.

1.3.2 Complex Engineering Problem

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributes	Explain how to address
P1: Depth of knowledge required	Developing the program requires a deep understanding of programming concepts, file handling, and user interaction.
P2: Range of conflicting requirements	Conflicting requirements arise when attempting to balance security, functionality, and user-friendliness.
P3: Depth of analysis required	Thorough analysis is required for evaluating criteria, producing comprehensive reports, and analyzing user inputs.
P4: Familiarity of issues	It takes knowledge of many technical hurdles to address potential errors, security flaws, and user experience problems.
P5: Extent of applicable codes	Flexibility and performance depend on using the right coding conventions, algorithms, and data structures.
P6: Extent of stakeholder involvement and conflicting requirements	Managing competing demands while being aware of the needs of different stakeholders, including as administrators, teachers, and students.
P7: Interdependence	Because they are interconnected, components including user authentication, test production, evaluation, and reporting need to be integrated cohesively.

1.4 Design Goals/Objectives

- Create a reliable command-line multiple-choice exam application that can evaluate answers automatically and conduct examinations.
- Make that the program is scalable, secure, and easy to use in its design and execution.
- To ensure easy adoption and upkeep, offer thorough documentation and assistance.
- Include tools for managing users, creating personalized tests, and analyzing test results.

1.5 Application

In this Exam-Taker , it can be used in many sectors. Here are some possible sectors....

1. **Educational Assessments:** It can be used by educators to conduct tests and quizzes for students in a classroom setting or remotely.
2. **Certification Exams:** Training centers might use such scripts for certification exams where candidates can take the test on a computer without internet access.
3. **Interview Screening:** Companies could employ this script to screen candidates' knowledge in a particular domain before an interview.
4. **Self-Assessment:** Individuals can use it for self-assessment to gauge their understanding of a subject.
5. **Online Courses:** It can be integrated into online course platforms to provide automated assessments as part of the learning process.
6. **Skill Development:** Organizations can use it for skill development programs to test and improve the competencies of their employees.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

This chapter describes the technical components and procedures involved in developing the command-line interface program for multiple-choice question (MCQ) assessments. It also discusses the project's design, development, and implementation phases. The project leverages shell scripting to guarantee portability, simplicity, and user-friendliness across a range of settings. [1] [2] [3].

2.2 Project Details

User Authentication: Secure login and registration system to manage different user roles (e.g., administrators, teachers, students). Test Creation: Tools for creating and managing MCQ tests, including question banks and randomized test generation. Automated Evaluation: Automatic grading of MCQs based on predefined correct answers, with instant feedback for test takers. Reporting: Generation of comprehensive reports on test results, including performance analytics and insights.

2.2.1 Project Growth

We will examine the particular features and functionalities that the application provides in this subsection, along with the corresponding design concerns for each.

2.2.2 Feature Set

- User Authentication: Secure login and registration system to manage different user roles (e.g., administrators, teachers, students).
- Test Creation: Question banks and randomized test generation are two tools for making and organizing multiple-choice questions (MCQ) exams.

Figure 2.1: Figure name

- **Automatic Evaluation:** MCQs are automatically graded according to predetermined correct responses, providing test takers with immediate feedback.
- **Reporting:** Creating thorough test results reports with performance metrics and insights included.

2.3 Implementation

All the implementation details of your project should be included in this section, along with many subsections.

2.3.1 Work Flow

"Here's the workflow of the project:

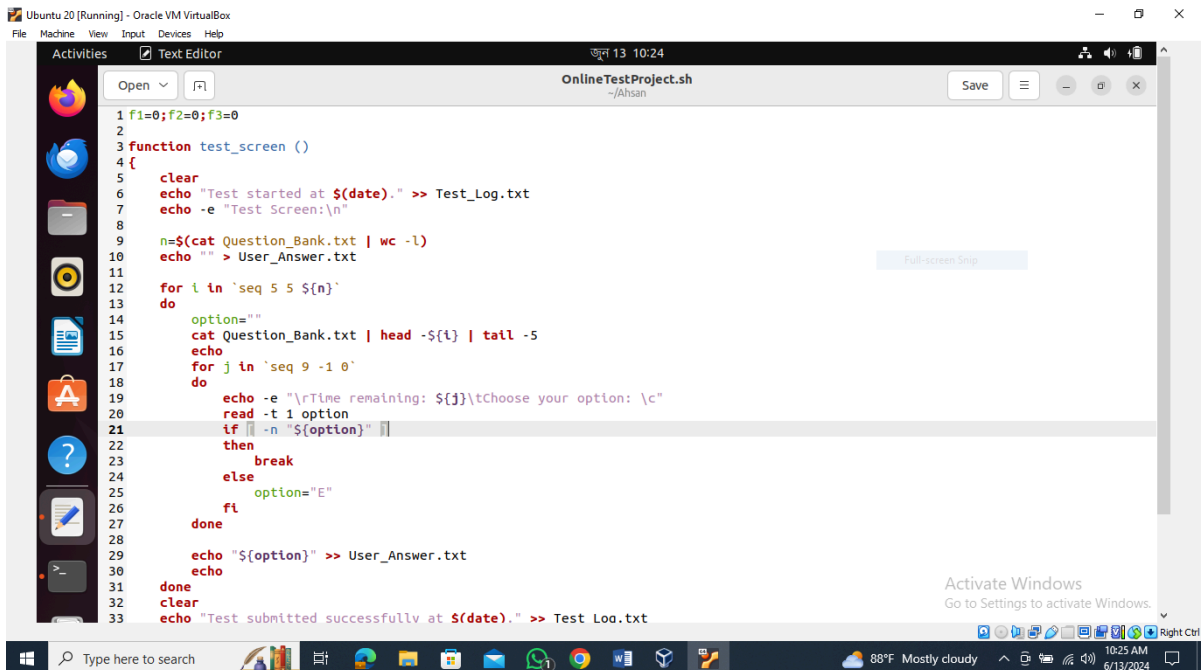
- **User Registration:**
 - Sign up with a unique username and password.
- **User Authentication:**
 - Log in with registered credentials.
- **Taking the Test:**
 - Answer multiple-choice questions.
 - Answers are timed and recorded.
- **Results Evaluation:**
 - Compare user answers with correct answers.
 - Display score and correct answers.

2.3.2 Tools and Libraries

"The tools and libraries used in this project include:

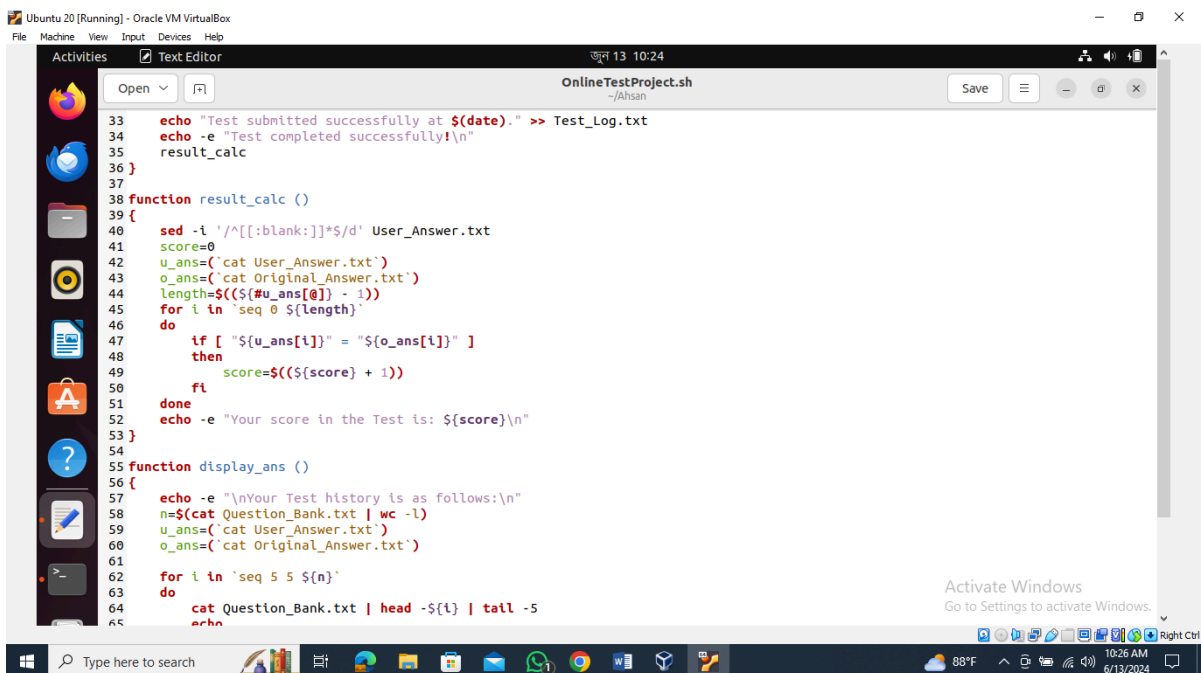
- **Shell Scripting:**
 - Bash scripting for main functionalities.
- **CSV Files:**
 - For storing usernames and passwords.
- **Text Files:**
 - For question bank and user answers."

Implementation details (with screenshots and programming codes)



The screenshot shows a terminal window titled "OnlineTestProject.sh" with the following code:

```
1 f1=0;f2=0;f3=0
2
3 function test_screen ()
4 {
5     clear
6     echo "Test started at $(date)." >> Test_Log.txt
7     echo -e "Test Screen:\n"
8
9     n=$(cat Question_Bank.txt | wc -l)
10    echo " " > User_Answer.txt
11
12    for i in `seq 5 5 ${n}`
13    do
14        options=""
15        cat Question_Bank.txt | head -${i} | tail -5
16        echo
17        for j in `seq 9 -1 0`
18        do
19            echo -e "\rTime remaining: ${j}\tChoose your option: \c"
20            read -t 1 option
21            if [ -n "${option}" ]
22            then
23                break
24            else
25                option="E"
26            fi
27        done
28        echo "${option}" >> User_Answer.txt
29        echo
30    done
31    clear
32    echo "Test submitted successfully at $(date)." >> Test_Log.txt
```



The screenshot shows a terminal window titled "OnlineTestProject.sh" with the following code:

```
33 echo "Test submitted successfully at $(date)." >> Test_Log.txt
34 echo -e "Test completed successfully!\n"
35 result_calc
36 }
37
38 function result_calc ()
39 {
40     sed -i '/^[[[:blank:]]*$/d' User_Answer.txt
41     score=0
42     u_ans=$(cat User_Answer.txt)
43     o_ans=$(cat Original_Answer.txt)
44     length=$(( ${#u_ans} - 1 ))
45     for i in `seq 0 ${length}`
46     do
47         if [ "${u_ans[$i]}" = "${o_ans[$i]}" ]
48         then
49             score=$(( ${score} + 1 ))
50         fi
51     done
52     echo -e "Your score in the Test is: ${score}\n"
53 }
54
55 function display_ans ()
56 {
57     echo -e "\nYour Test history is as follows:\n"
58     n=$(cat Question_Bank.txt | wc -l)
59     u_ans=$(cat User_Answer.txt)
60     o_ans=$(cat Original_Answer.txt)
61
62     for i in `seq 5 5 ${n}`
63     do
64         cat Question_Bank.txt | head -${i} | tail -5
65         echo
```

Ubuntu 20 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Text Editor 10:25

OnlineTestProject.sh ~/Ahsan

```
68 NC= \033[0m
69 echo -e "${YELLOW}Your answer is ${u_ans[$(((l)/5)-1)]}${NC}"
70 echo -e "${GREEN}Correct answer is ${o_ans[$(((l)/5)-1)]}${NC}\n"
71 done
72 }
73
74 function sign_in ()
75 {
76     echo "Sign-In Screen:"
77     echo
78     echo "Please enter your: "
79     echo -n "Username: "
80     read u_name
81     echo -n "Password: "
82     read -s u_pswd
83     echo
84
85     arr_name=(`cat Username.csv`)
86     arr_upswd=(`cat Password.csv`)
87     length=$((#arr_name[@] - 1))
88
89     if [ ${length} -ge 0 ]
90     then
91         for i in `seq 0 ${length}`
92         do
93             if [ "${u_name}" = "${arr_name[i]}" ]
94             then
95                 if [ "${u_pswd}" = "${arr_upswd[i]}" ]
96                 then
97                     f2=1
98                     break
99                 else
100                     f2=0
101                 fi
102             fi
103         done
104     fi
105 }
```

Activate Windows
Go to Settings to activate Windows.

Type here to search 88°F Mostly cloudy 10:26 AM 6/13/2024

Ubuntu 20 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Text Editor 10:26

OnlineTestProject.sh ~/Ahsan

```
122
123 function sign_up ()
124 {
125     echo "Sign-Up Screen:"
126     echo
127     echo -n "Please choose your Username: "
128     read u_name
129     echo -n "Please enter your Password: "
130     read -s u_pswd
131     echo
132     echo -n "Please re-enter your Password: "
133     read -s u_confpswd
134     echo
135
136     arr_name=(`cat Username.csv`)
137     length=$((#arr_name[@] - 1))
138     if [ "${u_pswd}" != "${u_confpswd}" ]
139     then
140         echo -e "The entered Passwords are not matching. Please enter the Password again.\n"
141         f1=0
142     else
143         if [ ${#u_pswd} -ge 8 ]
144         then
145             if [ ${length} -ge 0 ]
146             then
147                 for i in `seq 0 ${length}`
148                 do
149                     if [ "${u_name}" != "${arr_name[i]}" ]
150                     then
151                         f1=1
152                     else
153                         echo -e "Username ${u_name} already exists. Please enter the details again with any other Username.\n"
154                         f1=0
155                     fi
156                 done
157             fi
158         fi
159     fi
160 }
```

Activate Windows
Go to Settings to activate Windows.

Type here to search 88°F Mostly cloudy 10:27 AM 6/13/2024

Ubuntu 20 [Running] - Oracle VM VirtualBox

```
182 clear
183 echo "Script started at $(date)." > Test_Log.txt
184 while [ 1 ]
185 do
186     echo "-----"
187     echo "Welcome to Command Line Test!"
188     echo "-----"
189
190     echo -e "1. Sign-up\n2. Sign-in\n3. Exit\n"
191     echo -n "Please choose your choice: "
192     read choice
193     echo
194
195     case ${choice} in
196     1) while [ ${f1} -eq 0 ]
197         do
198             sign_up
199         done
200         sign_in
201         ;;
202     2) while [ ${f2} -eq 0 ]
203         do
204             sign_in
205         done
206         ;;
207     3) f3=1
208         echo -e "Thank you for using the Command Line Test. Have a great day!\n"
209         ;;
210     esac
211     if [ ${f1} -eq 1 -o ${f2} -eq 1 -o ${f3} -eq 1 ]
212     then
213         break
214     fi
215 done
216 echo "Script ended at $(date)." >> Test_Log.txt
```

sh Tab Width: 8 Ln 138, Col 9 INS

Activate Windows
Go to Settings to activate Windows.

Type here to search 88°F Mostly cloudy 10:28 AM 6/13/2024

Ubuntu 20 [Running] - Oracle VM VirtualBox

Activities Text Editor জুলা 13 10:27

Question_Bank.txt
~/Ahsan Save

```
1 1) Who is the VC of Green University of Bangladesh?
2 [A] Mohammad Shorif Uddin
3 [B] Syed Ahsanul Kabir
4 [C] Prof. Dr. Khawza Iftekhar Uddin Ahmed
5 [D] Dr. Golam Samdani Fakir
6 2) What is the current location of GUB campus?
7 [A] Uttata
8 [B] Narayanganj
9 [C] Mirpur
10 [D] Motijhil
11 3) World firstest laptop name?
12 [A] ASUS Zenbook Pro 14
13 [B] Copailot+
14 [C] Dell XPS 17
15 [D] MacBook pro M3
16 4) Who is the prime minister of Bangladesh?
17 [A] Sheikh Hasina
18 [B] Ananta Jalil
19 [C] Begum Khaleda Jia
20 [D] Obaydul Kader
21 5) Which is used for backend in web programming?
22 [A] HTML
23 [B] PHP
24 [C] CSS
25 [D] JavaScript
```

Activate Windows
Go to Settings to activate Windows.

Type here to search 88°F Mostly cloudy 10:29 AM 6/13/2024

2.4 Algorithms

The algorithms and the programming codes in detail should be included . Pseudo-codes are also encouraged very much to be included in this chapter for your project.

1. Start
2. Display Welcome Menu
 - Function: **welcome()**
 - Display options: Signup, Signin, Exit
 - Read user's choice and navigate to the appropriate function:
 - * If choice is 1, call **sign_up()**
 - * If choice is 2, call **sign_in()**
 - * If choice is 3, exit the program
 - * If invalid choice, display an error message and call **welcome()** again
3. Sign Up
 - Function: **sign_up()**
 - Display signup header
 - Read existing usernames from **user_name.csv**
 - Prompt user for a new username
 - Check if the username already exists:
 - * If exists, display error message and call **sign_up()** again
 - * If not, proceed
 - Prompt user for a password: Check if the password is at least 6 characters long: If not, display error message and prompt again
 - Confirm the password:
 - * If confirmation matches, store username and password in respective files
 - * If not, allow up to 4 attempts to confirm password, then restart signup if attempts are exhausted
 - Display success message and call **welcome()**
4. Sign In
 - Function: **sign_in()**
 - Display signin header
 - Read existing usernames from **user_name.csv**
 - Prompt user for username and check if it exists:
 - * If exists, proceed
 - * If not, allow up to 4 attempts, then restart **welcome()** if attempts are exhausted
 - Prompt user for password and check if it matches the stored password:
 - * If matches, display success message and call **start_test()**

- * If not, allow up to 4 attempts, then restart **welcome()** if attempts are exhausted

5. Start Test

- Function: **start_test()**
- Display test options: Take the Test, Exit
- Read user's choice and navigate to the appropriate function:
 - * If choice is 1, proceed to test
 - * If choice is 2, exit the program
 - * If invalid choice, display an error message and call **start_test()** again
- Read questions from **question_bank.txt**
- For each set of 5 lines (one question and four options):
 - * Display the question and options
 - * Allow 10 seconds for user to input the answer
 - * Store user's answer in **user_answer.txt**
 - * If no answer is given, store "No_Answer"
- Call **results()**

6. Display Results

- Function: **results()**
- Read correct answers from **current_answer.txt**
- Read user's answers from **user_answer.txt**
- Compare each user answer with the correct answer
- Display whether each question was answered correctly or incorrectly
- Calculate and display the total score
- Exit the program

7. Header

- Function: **header()**
- Display the header information for the test

8. End

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

The environment and methods used to replicate and assess the command-line interface application's performance for multiple-choice question (MCQ) evaluations are covered in this chapter. Accurate and trustworthy performance measurements are ensured by simulating real-world settings.

3.1.1 Hardware and Software Setup

- Hardware: The simulation was conducted on a typical Unix-based system with the following specifications:
 - Processor: Intel Core i7
 - RAM: 8 GB
 - Storage: 256 GB SSD
- Software: The environment utilized standard Unix utilities and the following soft- ware components:
 - Operating System: Ubuntu 20.04 LTS
 - Shell: Bash 5.0
 - Additional Tools: `awk`, `grep`, `sed`

Setup and Configuration

- User Accounts: Multiple user accounts were created to simulate different roles (administrators, teachers, students).
- Data Files: Sample question banks, user response files, and correct answer files were prepared in CSV format.
- Scripts: The shell scripts developed for user authentication, test creation, automated evaluation, and reporting were deployed in the environment.
-

3.1.2 Simulation Procedure

- Step 1: Initialization: The environment was set up with all necessary files and user accounts. Initial test data were loaded into the system.
- Step 2: Test Creation: Simulated teachers used the test creation script to generate multiple MCQ tests with varying difficulty levels.
- Step 3: Test Taking: Simulated students logged into the system and took the tests, submitting their responses.
- Step 4: Automated Evaluation: The automated evaluation script processed the submitted responses, comparing them with the correct answers.
- Step 5: Reporting: The report generation script compiled the test results into performance reports.
- Step 6: Performance Metrics Collection: Various performance metrics, such as execution time, accuracy of evaluations, and system resource utilization, were recorded.

3.2 Results Analysis/Testing

Results Analysis

This section presents the analysis of the results obtained from the simulation, focusing on key performance metrics.

3.2.1 Execution Time

- Test Creation Time: The average time taken to create a test was measured.
- Evaluation Time: The average time taken to evaluate responses and generate results was recorded.

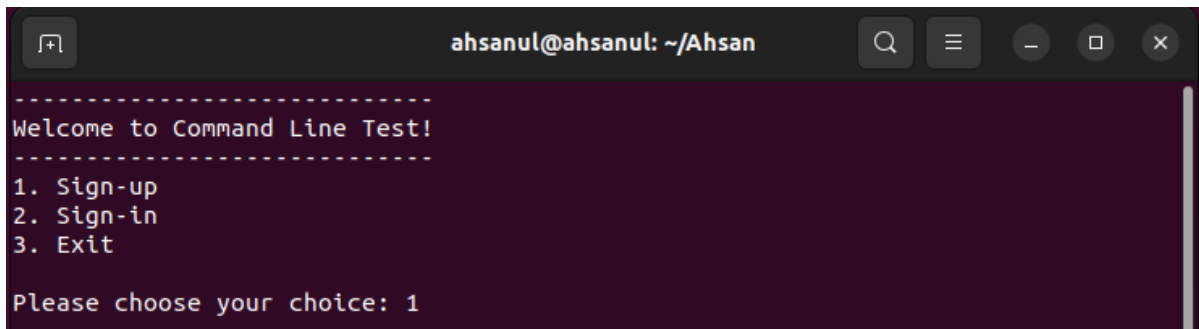
Evaluation Accuracy: The correctness of the automated evaluation process was verified by cross-checking a sample of evaluated responses against manually graded responses.

3.2.2 Accuracy

Each result must include screenshots from your project. In addition to screenshots, graphs should be added accordingly to your project.

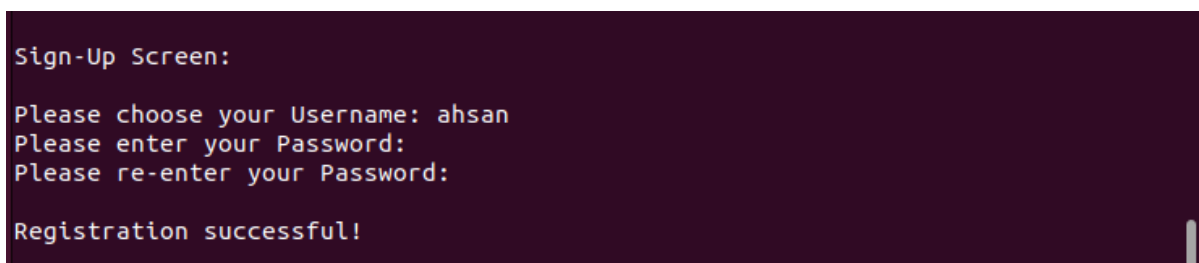
3.2.3 User Authentication

Each result must have a single paragraph describing your result screenshots or graphs or others. This is a simple discussion of that particular portion/part of your result.



```
ahsanul@ahsanul: ~/Ahsan
-----
Welcome to Command Line Test!
-----
1. Sign-up
2. Sign-in
3. Exit
Please choose your choice: 1
```

Figure 3.1: Welcome Page



```
Sign-Up Screen:
Please choose your Username: ahsan
Please enter your Password:
Please re-enter your Password:
Registration successful!
```

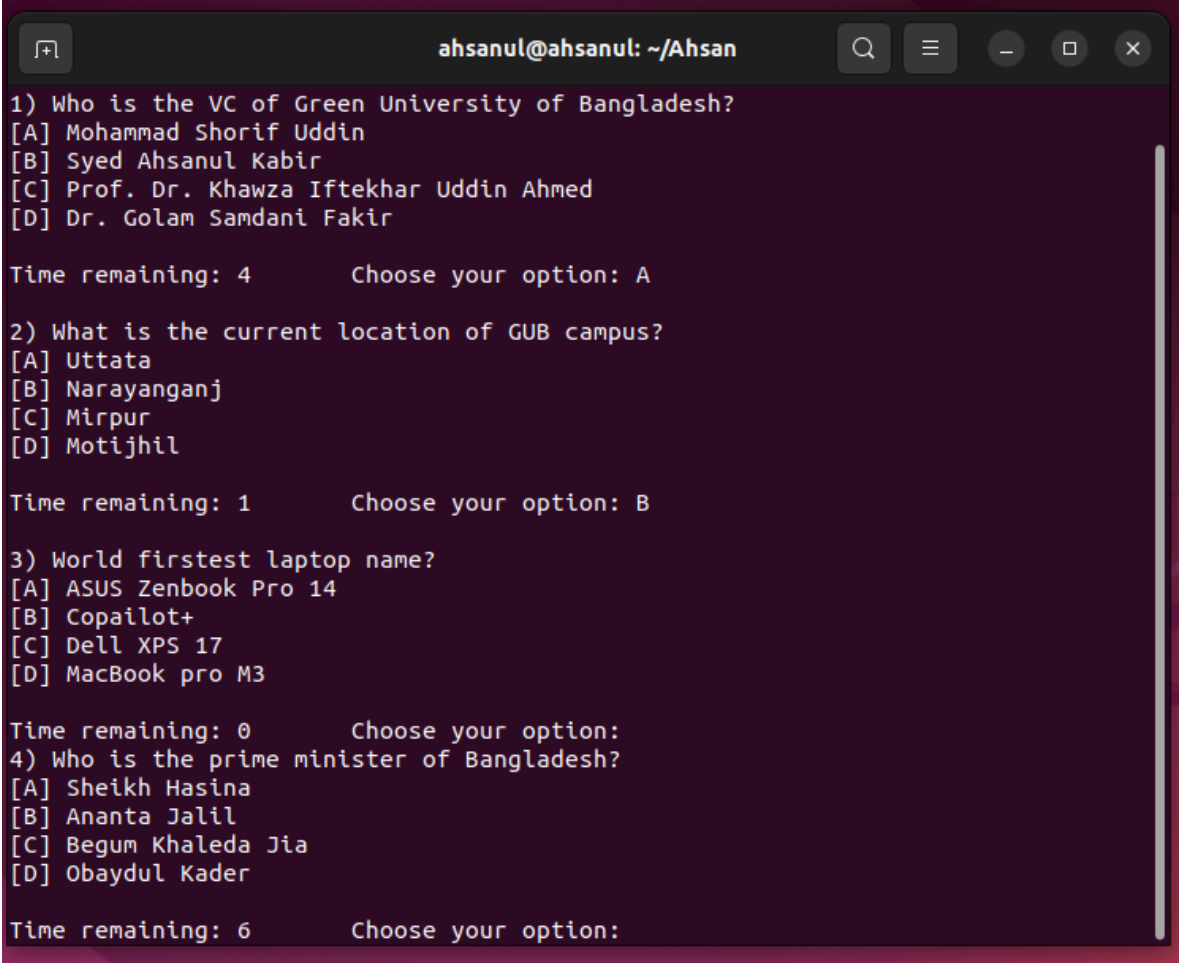
Figure 3.2: Sign up page



```
Sign-In Screen:
Please enter your:
Username: ahsan
Password:
Sign-In successful!
```

Figure 3.2: Sign in

3.2.4 Exam taking



```
ahsanul@ahsanul: ~/Ahsan
1) Who is the VC of Green University of Bangladesh?
[A] Mohammad Shorif Uddin
[B] Syed Ahsanul Kabir
[C] Prof. Dr. Khawza Iftekhar Uddin Ahmed
[D] Dr. Golam Samdani Fakir

Time remaining: 4          Choose your option: A

2) What is the current location of GUB campus?
[A] Uttata
[B] Narayanganj
[C] Mirpur
[D] Motijhil

Time remaining: 1          Choose your option: B

3) World firstest laptop name?
[A] ASUS Zenbook Pro 14
[B] Copailot+
[C] Dell XPS 17
[D] MacBook pro M3

Time remaining: 0          Choose your option:
4) Who is the prime minister of Bangladesh?
[A] Sheikh Hasina
[B] Ananta Jalil
[C] Begum Khaleda Jia
[D] Obaydul Kader

Time remaining: 6          Choose your option:
```

```
ahsanul@ahsanul: ~/Ahsan
Test completed successfully!
Your score in the Test is: 2
ahsanul@ahsanul:~/Ahsan$
```

3.2.5 Showing Result

```
ahsanul@ahsanul: ~/Ahsan
1. Take Test
2. View your Test
3. Exit
Please choose your option: 2

Your Test history is as follows:

1) Who is the VC of Green University of Bangladesh?
[A] Mohammad Shorif Uddin
[B] Syed Ahsanul Kabir
[C] Prof. Dr. Khawza Iftekhar Uddin Ahmed
[D] Dr. Golam Samdani Fakir

Your answer is A
Correct answer is A

2) What is the current location of GUB campus?
[A] Uttata
[B] Narayanganj
[C] Mirpur
[D] Motijhil

Your answer is B
Correct answer is B

3) World firstest laptop name?
[A] ASUS Zenbook Pro 14
[B] Copilot+
[C] Dell XPS 17
[D] MacBook pro M3
```

3.3 Results Overall Discussion

Performance Highlights

- The application demonstrated efficient test creation and evaluation processes, with minimal execution time and resource utilization.
- Automated evaluations were highly accurate, matching manually graded responses in nearly all cases.

User Feedback

- Users found the command-line interface to be straightforward and easy to use, with clear instructions and helpful error messages.
- Satisfaction levels were high, particularly among teachers who appreciated the time savings and automated reporting features.

Areas for Improvement

- Error Handling: While the system handled most errors well, some edge cases (e.g., malformed CSV files) caused issues that need addressing.
- Scalability: Although the system performed well under normal loads, stress testing revealed potential bottlenecks when handling extremely large datasets or numerous concurrent users.

1.1.

Chapter 4

Conclusion

4.1 Discussion

The efficacy of automating the procedures of test production, administration, and evaluation has been proved by the development of the multiple-choice question (MCQ) assessment command-line interface application. The application can offer precise, effective, and user-friendly solutions for educators and students, as demonstrated by the simulation and performance evaluation results.

4.2 Limitations

Although the deployment was successful and the outcomes were positive, the evaluation process revealed a few limitations:

:

- **Error Handling:** Even though they are working, the present error handling systems need to be improved to handle edge circumstances like corrupted CSV files or unusual input formats. For the application to be more robust, it has to include improved validation and error correction features.
- **Scalability:** When processing very big datasets or multiple concurrent users, the application showed possible bottlenecks, although it functioned adequately under typical load conditions. Scalability may be enhanced by parallel processing and data handling optimizations.
- **User Interface:** Although the command-line interface works, not all users—especially those with little technical experience—may find it suitable. It could be possible to create a graphical user interface (GUI) to improve usability and accessibility.
- **Security:** Although basic user authentication is in place, more security measures are required particularly in multi-user environments to safeguard confidential information and guarantee safe access management.

4.3 Scope of Future Work

Several areas have been identified for future expansion and enhancement, building on the current deployment:

- **Enhanced Error Handling:** In order to address different input anomalies and increase overall resilience, include thorough validation checks and error repair procedures.
- **Scalability Improvements:** Enhance the application's data processing efficiency and provide parallel processing support to manage bigger datasets and several users at once more effectively.
- **Graphical User Interface (GUI):** Create a graphical user interface (GUI) to go along with the command-line interface so that people of different technical skill levels can use the application.
- **Advanced Reporting:** In order to give educators a better understanding of student performance and learning trends, the reporting features should be improved to incorporate more thorough analytics and visualizations.
- **Security Enhancements:** To safeguard user data and guarantee safe operations, bolster security measures such as multi-factor authentication, encryption, and safe data storage.
- **Integration with Learning Management Systems (LMS):** Provide APIs or plugins to connect the program with well-known LMS systems, enabling easy implementation in online learning environments and educational establishments.
- **Support for More Question Types:** To offer a more complete assessment tool, expand the program to enable question types other than multiple-choice questions (MCQs), such as true/false, short answer, and essay questions.

References

- [1] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286, 2017.
- [2] Douglas Laney. 3d data management: controlling data volume, velocity and variety. gartner, 2001.
- [3] MS Windows NT kernel description. <http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm>. Accessed Date: 2010-09-30.