

**CS2006**

Operating Systems

**Semester Project 1**

**Submitted to:** Hassan Ahmed

**Name:** Fatima Nasir, Meerab Awais

**Section:** 4A

**Roll no#** 23L-0714, 23F-0515

**Description:**

The student management system is designed in bash scripting to manage student’s grades and CGPA. It allows a single teacher to manage 20 students by adding them to the files, deleting them from the files, assigning marks to them, calculate their grades and generate report while students can look into their results, i.e. CGPA and grades assigned by the teacher.

**Code:**

###########################################################################

#Student Management System : Teachers can manage student’s data by adding a student, deleting a student, assigning marks and grades to them. And Students can also check their grades and CGPA.

###########################################################################

#!/bin/bash

Sdatafile="students.txt" # file to store student data

#............................................................................................

# check if a student exists by roll number

exists() {

while IFS=, read -r id \_; do

if [ "$id" == "$1" ]; then

return 0 # student exists

fi

done < "$Sdatafile"

return 1 # student does not exist

}

#............................................................................................

# add a new student

addstudent() {

echo "enter roll number:"

read roll

if exists "$roll"; then

echo "student with roll no $roll already exists!"

return

fi

echo "enter name:"

read name

echo "enter section:"

read section

echo "$roll,$name,$section,0,0,F" >> "$Sdatafile" # add student to the file

echo "student added successfully!"

}

#...........................................................................................

# function to delete a student

delete() {

echo "enter roll number to delete:" # ask user for roll number

read roll

if ! exists "$roll"; then # check if student exists

echo "student not found!" # show message if student does not exist

return

fi

> temp.txt # create an empty temporary file

# read file line by line and remove the student with matching roll number

while IFS=, read -r id rest; do

if [ "$id" != "$roll" ]; then # check if roll number is not the one to delete

echo "$id,$rest" >> temp.txt # write the line to temp file if not deleting

fi

done < "$Sdatafile"

mv temp.txt "$Sdatafile" # replace original file with updated data

echo "student deleted successfully!"

}

#.............................................................................................

# function to assign marks to a student

assignmarks() {

echo "enter roll number:"

read roll

if ! exists "$roll"; then # check if student exists

echo "student not found!" # show message if student is not found

return

fi

echo "enter marks:"

read marks

> temp.txt # create an empty temporary file

# read file line by line and update marks for the matching student

while IFS=, read -r id name section oldmarks cgpa grade; do

if [ "$id" == "$roll" ]; then # check if current student is the one to update

echo "$id,$name,$section,$marks,$cgpa,$grade" # update marks

else

echo "$id,$name,$section,$oldmarks,$cgpa,$grade" # keep other students unchanged

fi

done < "$Sdatafile" > temp.txt

mv temp.txt "$Sdatafile" # replace original file with updated data

echo "marks assigned successfully!"

}

#.............................................................................................

# function to calculate grades based on marks

calculategrade() {

> temp.txt # creating an empty temporary file

# read each student's data and assign grades based on marks

while IFS=, read -r id name section marks cgpa grade; do

if [ "$marks" -ge 90 ]; then grade="A+" # if marks are 90 or above, assign A+

elif [ "$marks" -ge 80 ]; then grade="A" # if marks are 80 or above, assign A

elif [ "$marks" -ge 75 ]; then grade="B+" # if marks are 75 or above, assign B+

elif [ "$marks" -ge 70 ]; then grade="B" # if marks are 70 or above, assign B

elif [ "$marks" -ge 65 ]; then grade="C+" # if marks are 65 or above, assign C+

elif [ "$marks" -ge 60 ]; then grade="C" # if marks are 60 or above, assign C

elif [ "$marks" -ge 50 ]; then grade="D" # if marks are 50 or above, assign D

else grade="F"; fi # if marks are below 50, assign F

echo "$id,$name,$section,$marks,$cgpa,$grade" # update the record with the new grade

done < "$Sdatafile" > temp.txt

mv temp.txt "$Sdatafile" # replace the original file with updated data

echo "grades calculated!"

}

#.............................................................................................

# calculate cgpa based on grades

calculatecgpa() {

> temp.txt

while IFS=, read -r id name section marks cgpa grade; do

case "$grade" in #setting cgpa based on grades

A+) new\_cgpa=4.0 ;;

A) new\_cgpa=3.6 ;;

B+) new\_cgpa=3.3 ;;

B) new\_cgpa=3.0 ;;

C+) new\_cgpa=2.5 ;;

C) new\_cgpa=2.0 ;;

D) new\_cgpa=1.5 ;;

F) new\_cgpa=0.0 ;;

esac

echo "$id,$name,$section,$marks,$new\_cgpa,$grade" #update record with new cgpa

done < "$Sdatafile" > temp.txt

mv temp.txt "$Sdatafile" #replace original file with updated one

echo "cgpa calculated!"

}

#..........................................................................................

#update student details function

update() {

echo "enter roll number to update:"

read roll

if ! exists "$roll"; then #checking for existence

echo "student not found!"

return

fi

#enter new details

echo "enter new name:"

read newname

echo "enter new section:"

read newsection

echo "enter new marks:"

read newmarks

> temp.txt

# read each student's data and update details if roll number matches

while IFS=, read -r id name section marks cgpa grade; do

if [ "$id" == "$roll" ]; then

echo "$id,$newname,$newsection,$newmarks,$cgpa,$grade"

else

echo "$id,$name,$section,$marks,$cgpa,$grade" #keep existing data for other students

fi

done < "$Sdatafile" > temp.txt

mv temp.txt "$Sdatafile"

echo "student details updated successfully!"

}

#...................................................................................

# generate reports function

generatereport() {

#part 1

echo "All students:"

cat "$Sdatafile"

#part 2

echo -e "\nPassed students (cgpa >= 2.0):"

while IFS=, read -r id name section marks cgpa grade; do

if (( $(echo "$cgpa >= 2.0" | bc -l) )); then echo "$id,$name,$section,$marks,$cgpa,$grade"; fi

done < "$Sdatafile"

#part 3

echo -e "\nFailed students (cgpa < 2.0):"

while IFS=, read -r id name section marks cgpa grade; do

if (( $(echo "$cgpa < 2.0" | bc -l) )); then echo "$id,$name,$section,$marks,$cgpa,$grade"; fi

done < "$Sdatafile"

#part 4

echo -e "\nSorted list (ascending by cgpa):"

sort -t, -k5 -n "$Sdatafile"

#part 5

echo -e "\nSorted list (descending by cgpa):"

sort -t, -k5 -nr "$Sdatafile"

}

#...............................................................................................

#function for student login

studentlogin() {

echo "enter roll number:"

read roll

if ! exists "$roll"; then

echo "student not found!"

return

fi

#student menu ,options for them

echo "1. view grades"

echo "2. view cgpa"

echo "choose an option:"

read choice

while IFS=, read -r id name section marks cgpa grade; do

if [ "$id" == "$roll" ]; then

[ "$choice" == "1" ] && echo "grade: $grade"

[ "$choice" == "2" ] && echo "cgpa: $cgpa"

fi

done < "$Sdatafile"

}

#.............................................................................................

#function for teacher login and teacher functionalities

teacherlogin() {

echo "Enter teacher password:"

read -s password

if [ "$password" != "fastcfd" ]; then #password check

echo "incorrect password!"

return

fi

while true; do

#teacher related functions

echo "Teacher menu:"

echo "1. add student"

echo "2. delete student"

echo "3. assign marks"

echo "4. update student details"

echo "5. calculate grades"

echo "6. calculate cgpa"

echo "7. generate report"

echo "8. logout"

echo "Enter choice:"

read choice

case "$choice" in #each case has its own function

1) addstudent ;;

2) delete ;;

3) assignmarks ;;

4) update ;;

5) calculategrade ;;

6) calculatecgpa ;;

7) generatereport ;;

8) return ;;

\*) echo "invalid choice!" ;;

esac

done

}

#.............................................................................................

#main part from where the output starts

while true; do

echo ".....Welcome to Student Management System......"

echo "1. Student login"

echo "2. Teacher login"

echo "3. Exit"

echo "Enter your Choice:"

read choice

case "$choice" in

1) studentlogin ;; #function calling for student

2) teacherlogin ;; #function calling for teacher

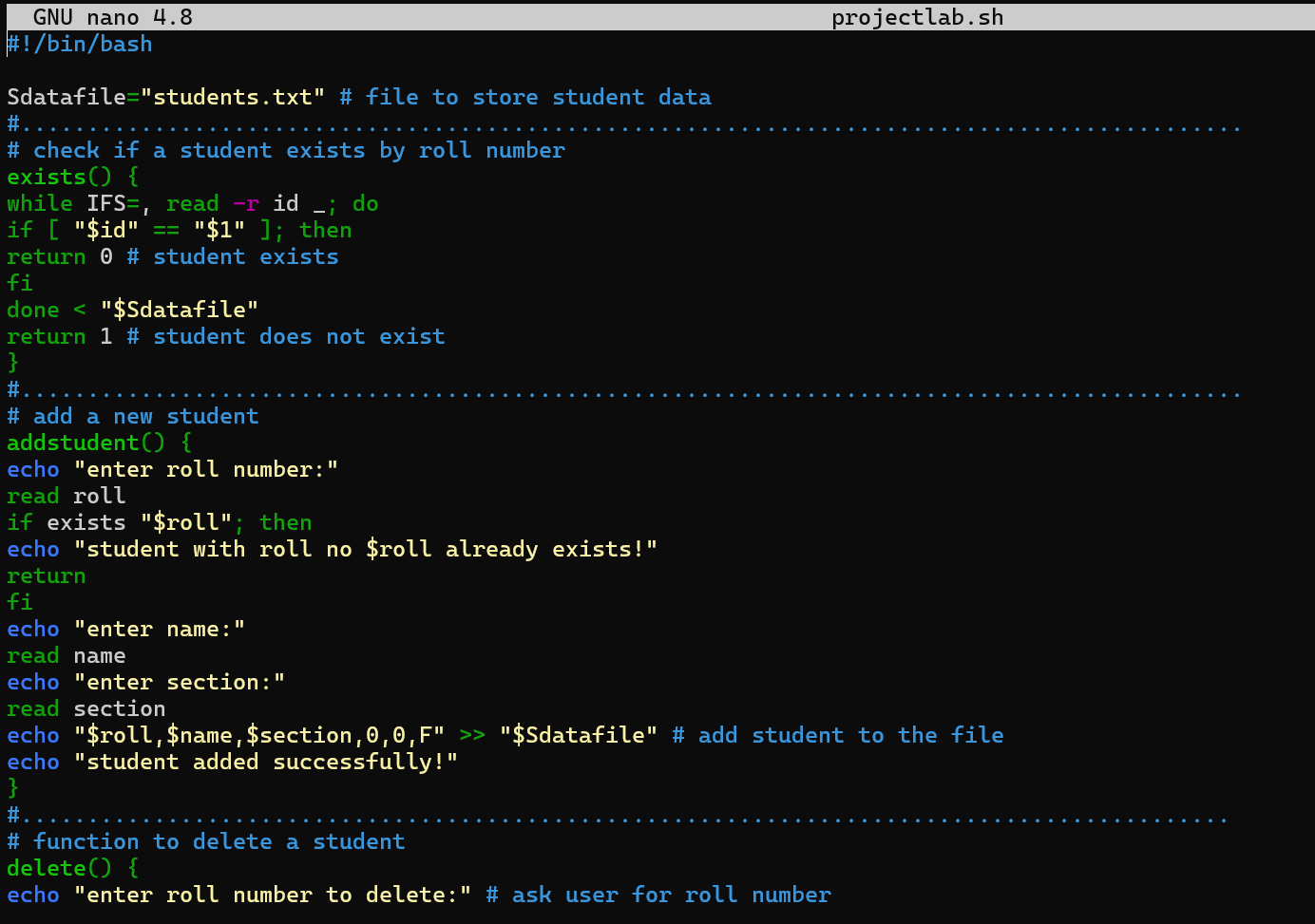
3) exit ;;

\*) echo "invalid choice!" ;;

esac

done

**Code screenshots:**

****

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer program

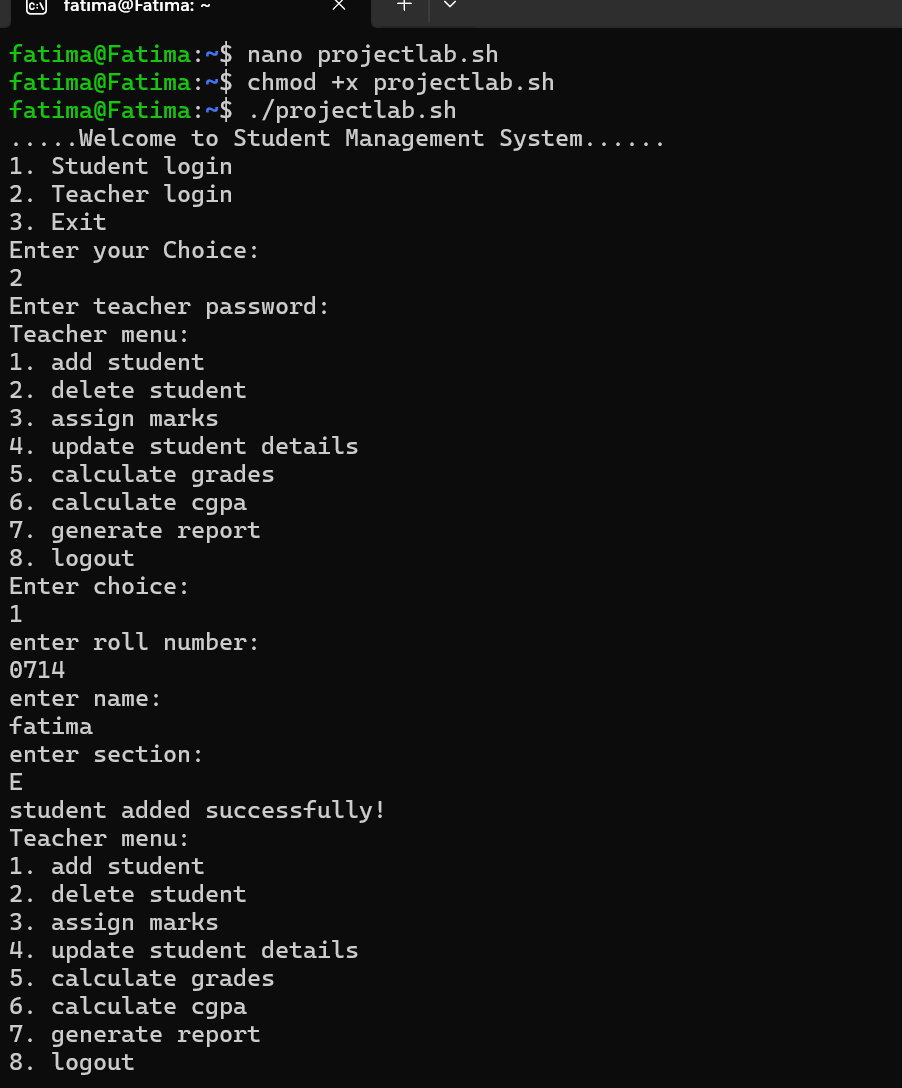
AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.Outputs:**

****

**A screenshot of a computer program

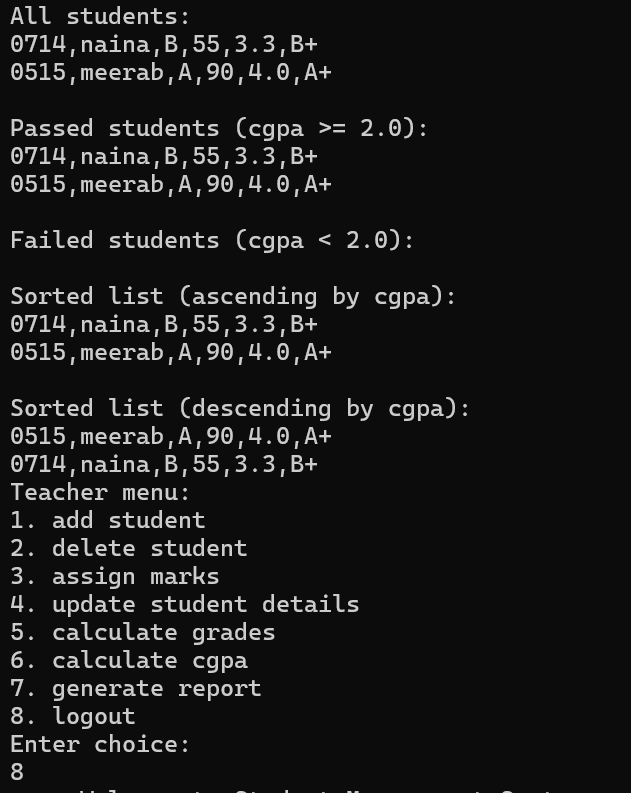
AI-generated content may be incorrect.**

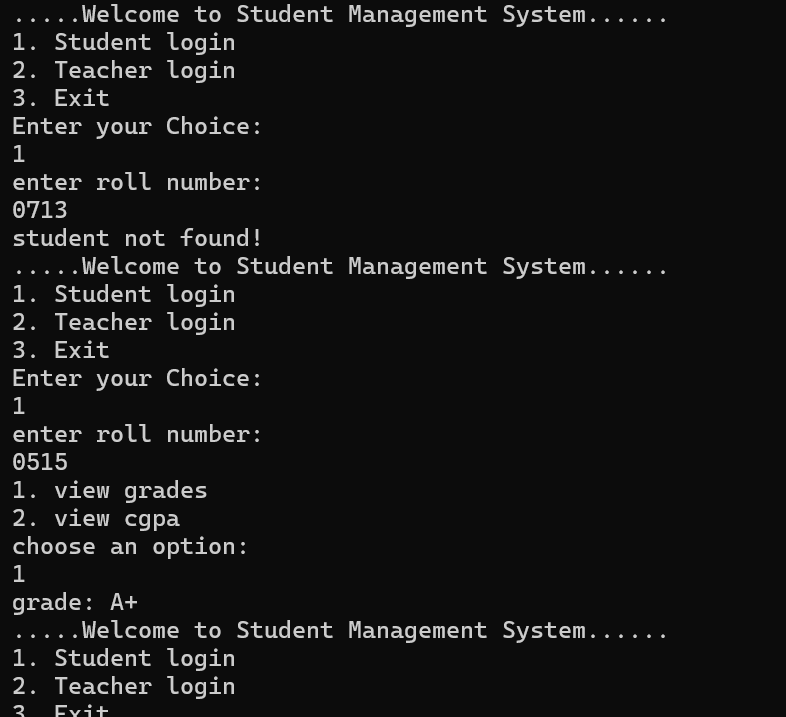
**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

****

****

**In the file:**

**A black background with white text

AI-generated content may be incorrect.**

**Error handling:**

**A screen shot of a computer

AI-generated content may be incorrect.**