|  |  |
| --- | --- |
|  |  |
| OS Project |  |
|  |  |
|  | 23F-0874Hasan Zubair |
|  | 23F-0505Hamza kamran |

**Code:**

#!/bin/bash

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

# Student Management System (SMS) - Bash Script

#

# Description:

#   A simple command-line system for teachers to manage student records and

#   for students to view their grades. Data is stored in text files.

#

# Key Features:

#   - Teacher: Add/delete students, update marks, calculate grades.

#   - Student: View grades using Roll No and password.

#   - Auto-saves data to students.txt and student\_logins.txt.

#   - Prevents duplicate roll numbers.

#   - Teachers set custom passwords for students.

#

# Usage:

#   1. Run the script: ./sms.sh

#   2. Default teacher login: admin / admin123

#   3. Teachers set student passwords during enrollment.

#

# File Structure:

#   - students.txt: Stores student records (Roll No, Name, Marks, Grade)

#   - student\_logins.txt: Stores student credentials (Roll No, Password)

#   - teachers.txt: Stores teacher credentials (Username, Password)

#

# Note: For first-time use, the script auto-creates a default teacher account.

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

# Files

STUDENTS\_FILE="students.txt"

TEACHERS\_FILE="teachers.txt"

LOGINS\_FILE="student\_logins.txt"

# Create files if they don't exist

touch "$STUDENTS\_FILE" "$TEACHERS\_FILE" "$LOGINS\_FILE"

# ====================== TEACHER FUNCTIONS ======================

teacher\_login() {

    echo -n "Teacher Username: "

    read username

    echo -n "Teacher Password: "

    read -s password

    echo

    # Check if teacher exists

    found=0

    while IFS=, read -r user pass; do

        if [ "$user" = "$username" ] && [ "$pass" = "$password" ]; then

            found=1

            break

        fi

    done < "$TEACHERS\_FILE"

    if [ $found -eq 1 ]; then

        teacher\_menu

    else

        echo "Invalid login!"

    fi

}

add\_student() {

    echo -e "\n--- Add Student ---"

    # Check for existing roll number

    while true; do

        echo -n "Roll No: "

        read roll\_no

        # Check if roll number already exists

        exists=0

        while IFS=, read -r r\_no name marks; do

            if [ "$r\_no" = "$roll\_no" ]; then

                exists=1

                break

            fi

        done < "$STUDENTS\_FILE"

        if [ $exists -eq 1 ]; then

            echo "Error: Roll No $roll\_no already exists! Please enter a different Roll No."

        else

            break

        fi

    done

    echo -n "Name: "

    read name

    echo -n "Marks: "

    read marks

    # Set password with validation

    while true; do

        echo -n "Set password for $name: "

        read -s password

        echo

        if [ -z "$password" ]; then

            echo "Password cannot be empty!"

        else

            break

        fi

    done

    # Save to students.txt

    echo "$roll\_no,$name,$marks" >> "$STUDENTS\_FILE"

    # Save to logins file

    echo "$roll\_no,$password,$roll\_no" >> "$LOGINS\_FILE"

    echo "Student added successfully!"

}

delete\_student() {

    echo -n "Enter Roll No to delete: "

    read roll\_no

    # Create a temp file without the student

    temp\_file=$(mktemp)

    while IFS=, read -r r\_no name marks; do

        if [ "$r\_no" != "$roll\_no" ]; then

            echo "$r\_no,$name,$marks" >> "$temp\_file"

        fi

    done < "$STUDENTS\_FILE"

    mv "$temp\_file" "$STUDENTS\_FILE"

    # Also delete from logins

    temp\_file=$(mktemp)

    while IFS=, read -r user pass r\_no; do

        if [ "$r\_no" != "$roll\_no" ]; then

            echo "$user,$pass,$r\_no" >> "$temp\_file"

        fi

    done < "$LOGINS\_FILE"

    mv "$temp\_file" "$LOGINS\_FILE"

    echo "Student deleted!"

}

assign\_marks() {

    echo -n "Enter Roll No to assign marks: "

    read roll\_no

    echo -n "Enter new marks: "

    read marks

    local input\_file="$STUDENTS\_FILE"

    local output\_file="temp"

    > "$output\_file"

    while IFS=',' read -r id name current\_marks

    do

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

            echo "$id,$name,$marks" >> "$output\_file"

        else

            echo "$id,$name,$current\_marks" >> "$output\_file"

        fi

    done < "$input\_file"

    mv "$output\_file" "$input\_file"

    echo "Marks updated."

}

calculate\_cgpa() {

    local output\_file="temp"

    local input\_file="$STUDENTS\_FILE"

    > "$output\_file"  # Empty or create temp file

    while IFS=',' read -r r\_no name marks grade gpa

    do

        gpa=0.0

        if (( marks >= 85 )); then

            gpa=4.0

        elif (( marks >= 80 )); then

            gpa=3.7

        elif (( marks >= 75 )); then

            gpa=3.3

        elif (( marks >= 70 )); then

            gpa=3.0

        elif (( marks >= 65 )); then

            gpa=2.7

        elif (( marks >= 60 )); then

            gpa=2.3

        elif (( marks >= 55 )); then

            gpa=2.0

        elif (( marks >= 50 )); then

            gpa=1.7

        elif (( marks >= 45 )); then

            gpa=1.0

        else

            gpa=0.0

        fi

        echo "$r\_no,$name,$marks,$grade,$gpa" >> "$output\_file"

    done < "$input\_file"

    mv "$output\_file" "$input\_file"

    echo "CGPA calculated."

}

calculate\_grades() {

    temp\_file=$(mktemp)

    while IFS=, read -r roll\_no name marks grade; do

        if [ -z "$marks" ]; then

            grade="N/A"

        elif [ "$marks" -ge 85 ]; then

            grade="A"

        elif [ "$marks" -ge 80 ]; then

            grade="A-"

        elif [ "$marks" -ge 75 ]; then

            grade="B+"

        elif [ "$marks" -ge 70 ]; then

            grade="B"

        elif [ "$marks" -ge 65 ]; then

            grade="B-"

        elif [ "$marks" -ge 60 ]; then

            grade="C+"

        elif [ "$marks" -ge 55 ]; then

            grade="C"

        elif [ "$marks" -ge 50 ]; then

            grade="C-"

        elif [ "$marks" -ge 45 ]; then

            grade="D+"

        else

            grade="F"

        fi

        echo "$roll\_no,$name,$marks,$grade" >> "$temp\_file"

    done < "$STUDENTS\_FILE"

    mv "$temp\_file" "$STUDENTS\_FILE"

    echo "Grades calculated!"

    calculate\_cgpa

}

sort\_by\_cgpa() {

    echo -e "\n--- Students Sorted by CGPA (Highest First) ---"

    echo "RollNo | Name   | Marks | Grade | GPA"

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

    # Sort numerically by GPA (5th field) in reverse order

    sort -t',' -k5,5 -nr "$STUDENTS\_FILE" | while IFS=',' read -r roll\_no name marks grade gpa

    do

        printf "%-7s| %-6s | %-5s | %-5s | %-4s\n" "$roll\_no" "$name" "$marks" "$grade" "$gpa"

    done

}

show\_passing\_students() {

    echo "Good Students Who Passed:"

    echo "-----------------------"

    # Read the file line by line

    while IFS=',' read -r roll name marks grade gpa

    do

        # Check if grade exists and isn't F

        if [ "$grade" != "F" ] && [ -n "$grade" ]; then

            echo "$roll | $name | $marks | $grade | $gpa"

        fi

    done < "$STUDENTS\_FILE"

    echo "-----------------------"

}

show\_failing\_students() {

    echo "Bad Students Who Failed:"

    echo "----------------------"

    # Read the file line by line

    while IFS=',' read -r roll name marks grade gpa

    do

        # Check if grade is F

        if [ "$grade" = "F" ]; then

            echo "$roll | $name | $marks | $grade | $gpa"

        fi

    done < "$STUDENTS\_FILE"

    echo "----------------------"

}

view\_students() {

    echo -e "\n--- Student List ---"

    echo "Roll No | Name       | Marks | Grade | GPA"

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

    while IFS=, read -r roll\_no name marks grade gpa; do

        # If GPA doesn't exist, show "N/A"

        [ -z "$gpa" ] && gpa="N/A"

        printf "%-8s| %-10s | %-5s | %-5s | %-4s\n" "$roll\_no" "$name" "$marks" "$grade" "$gpa"

    done < "$STUDENTS\_FILE"

}

# ====================== STUDENT FUNCTIONS ======================

student\_login() {

    echo -n "Student Username (Roll No): "

    read username

    echo -n "Password: "

    read -s password

    echo

    found=0

    while IFS=, read -r user pass r\_no; do

        if [ "$user" = "$username" ] && [ "$pass" = "$password" ]; then

            found=1

            break

        fi

    done < "$LOGINS\_FILE"

    if [ $found -eq 1 ]; then

        student\_menu "$r\_no"

    else

        echo "Invalid login!"

    fi

}

view\_grades() {

    roll\_no=$1

    while IFS=, read -r r\_no name marks grade; do

        if [ "$r\_no" = "$roll\_no" ]; then

            echo -e "\n--- Your Grades ---"

            echo "Name: $name"

            echo "Marks: $marks"

            echo "Grade: $grade"

            return

        fi

    done < "$STUDENTS\_FILE"

    echo "Record not found!"

}

# ====================== MENUS ======================

teacher\_menu() {

   while true; do

    echo -e "\n--- Teacher Menu ---"

    echo "1. Add new student"

    echo "2. Remove student"

    echo "3. Calculate grades"

    echo "4. Show all students"

    echo "5. Give marks to student"

    echo "6. Calculate GPA"

    echo "7. Show passing students"

    echo "8. Show failing students"

    echo "9. Sort by GPA"

    echo "10. Exit"

    echo -n "What do you want to do? (1-10): "

    read choice

    case $choice in

        1) add\_student ;;

        2) delete\_student ;;

        3) calculate\_grades ;;

        4) view\_students ;;

        5) assign\_marks ;;

        6) calculate\_cgpa ;;

        7) show\_passing\_students ;;

        8) show\_failing\_students ;;

        9) sort\_by\_cgpa ;;

        10) break ;;

        \*) echo "Oops! Try again with number 1-10" ;;

    esac

done

}

student\_menu() {

    roll\_no=$1

    while true; do

        echo -e "\n--- Student Menu ---"

        echo "1. View Grades"

        echo "2. Logout"

        echo -n "Choose an option: "

        read choice

        case $choice in

            1) view\_grades "$roll\_no" ;;

            2) break ;;

            \*) echo "Invalid option!" ;;

        esac

    done

}

main\_menu() {

    # Create default teacher if file is empty

    if [ ! -s "$TEACHERS\_FILE" ]; then

        echo "admin,admin123" > "$TEACHERS\_FILE"

    fi

    while true; do

        echo -e "\n--- Student Management System ---"

        echo "1. Teacher Login"

        echo "2. Student Login"

        echo "3. Exit"

        echo -n "Choose an option: "

        read choice

        case $choice in

            1) teacher\_login ;;

            2) student\_login ;;

            3) exit 0 ;;

            \*) echo "Invalid option!" ;;

        esac

    done

}

# Start the program

main\_menu

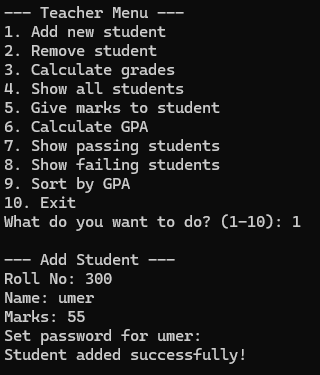
Outputs:

1-Login to teacher:

A screen shot of a computer

AI-generated content may be incorrect.

2-Adding a student:



3-Calculate grades:

A screen shot of a computer

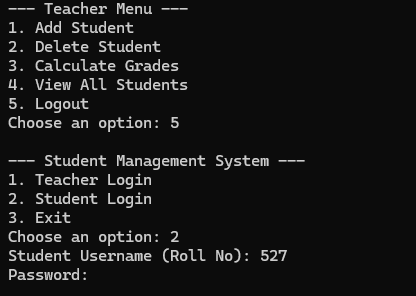
AI-generated content may be incorrect.

4-Print student list:

A black and white screen with white text

AI-generated content may be incorrect.

5-Logout from teacher and login to student :



6-View grades for student:

