

Operating system

Project mini



April 3, 2025

Azan Faisal,Arsal Hassan

Fast

Code:

#!/bin/bash

TEACHER\_FILE="teacher\_credentials.txt"

STUDENT\_FILE="student\_records.txt"

LOGIN\_FILE="login\_status.txt"

initialize\_files() {

if [ ! -f "$TEACHER\_FILE" ]; then

echo "admin:password123" > "$TEACHER\_FILE"

fi

if [ ! -f "$STUDENT\_FILE" ]; then

touch "$STUDENT\_FILE"

fi

if [ ! -f "$LOGIN\_FILE" ]; then

touch "$LOGIN\_FILE"

fi

}

teacher\_login() {

clear

echo "===== TEACHER LOGIN ====="

echo -n "Username: "

read username

echo -n "Password: "

read -s password

echo

stored\_cred=$(grep "^$username:" "$TEACHER\_FILE" | cut -d: -f2)

if [ "$password" = "$stored\_cred" ]; then

echo "teacher" > "$LOGIN\_FILE"

echo "$username" >> "$LOGIN\_FILE"

echo "Login successful!"

sleep 1

teacher\_menu

else

echo "Invalid credentials!"

sleep 1

main\_menu

fi

}

teacher\_menu() {

while true; do

clear

echo "===== TEACHER MENU ====="

echo "Logged in as: $(sed -n '2p' "$LOGIN\_FILE")"

echo "1. Add Student"

echo "2. Update Student Record"

echo "3. Delete Student"

echo "4. Assign Marks"

echo "5. Calculate Grades and CGPA"

echo "6. View All Students"

echo "7. View Passed Students"

echo "8. View Failed Students"

echo "9. Sort Students (Ascending by CGPA)"

echo "10. Sort Students (Descending by CGPA)"

echo "11. Logout"

echo "12. Exit System"

read -p "Enter your choice: " choice

case $choice in

1) add\_student ;;

2) update\_student ;;

3) delete\_student ;;

4) assign\_marks ;;

5) calculate\_grades\_cgpa ;;

6) view\_all\_students ;;

7) view\_passed\_students ;;

8) view\_failed\_students ;;

9) sort\_students\_ascending ;;

10) sort\_students\_descending ;;

11) logout ;;

12) exit\_system ;;

\*) echo "Invalid choice!"; sleep 1 ;;

esac

done

}

add\_student() {

clear

echo "===== ADD STUDENT ====="

student\_count=$(wc -l < "$STUDENT\_FILE")

if [ "$student\_count" -ge 20 ]; then

echo "Maximum of 20 students reached!"

sleep 1

return

fi

while true; do

read -p "Enter Roll No (must be unique): " roll\_no

if grep -q "^$roll\_no:" "$STUDENT\_FILE"; then

echo "Roll No already exists!"

else

break

fi

done

read -p "Enter Name: " name

read -p "Enter Marks (out of 100): " marks

while [[ ! "$marks" =~ ^[0-9]+$ ]] || [ "$marks" -lt 0 ] || [ "$marks" -gt 100 ]; do

read -p "Invalid marks! Enter Marks (0-100): " marks

done

echo "$roll\_no:$name:$marks:N/A:0.00" >> "$STUDENT\_FILE"

echo "Student added successfully!"

sleep 1

}

update\_student() {

clear

echo "===== UPDATE STUDENT ====="

view\_all\_students

echo

read -p "Enter Roll No of student to update: " roll\_no

if grep -q "^$roll\_no:" "$STUDENT\_FILE"; then

current\_data=$(grep "^$roll\_no:" "$STUDENT\_FILE")

IFS=':' read -ra data <<< "$current\_data"

echo "Current Name: ${data[1]}"

read -p "Enter new Name (leave blank to keep current): " new\_name

if [ -z "$new\_name" ]; then

new\_name="${data[1]}"

fi

echo "Current Marks: ${data[2]}"

read -p "Enter new Marks (leave blank to keep current): " new\_marks

if [ -z "$new\_marks" ]; then

new\_marks="${data[2]}"

else

while [[ ! "$new\_marks" =~ ^[0-9]+$ ]] || [ "$new\_marks" -lt 0 ] || [ "$new\_marks" -gt 100 ]; do

read -p "Invalid marks! Enter Marks (0-100): " new\_marks

done

fi

sed -i "/^$roll\_no:/d" "$STUDENT\_FILE"

echo "$roll\_no:$new\_name:$new\_marks:${data[3]}:${data[4]}" >> "$STUDENT\_FILE"

echo "Student record updated!"

else

echo "Student not found!"

fi

sleep 1

}

delete\_student() {

clear

echo "===== DELETE STUDENT ====="

view\_all\_students

echo

read -p "Enter Roll No of student to delete: " roll\_no

if grep -q "^$roll\_no:" "$STUDENT\_FILE"; then

sed -i "/^$roll\_no:/d" "$STUDENT\_FILE"

echo "Student deleted successfully!"

else

echo "Student not found!"

fi

sleep 1

}

assign\_marks() {

clear

echo "===== ASSIGN MARKS ====="

view\_all\_students

echo

read -p "Enter Roll No of student: " roll\_no

if grep -q "^$roll\_no:" "$STUDENT\_FILE"; then

current\_data=$(grep "^$roll\_no:" "$STUDENT\_FILE")

IFS=':' read -ra data <<< "$current\_data"

echo "Current Marks: ${data[2]}"

read -p "Enter new Marks: " new\_marks

while [[ ! "$new\_marks" =~ ^[0-9]+$ ]] || [ "$new\_marks" -lt 0 ] || [ "$new\_marks" -gt 100 ]; do

read -p "Invalid marks! Enter Marks (0-100): " new\_marks

done

sed -i "/^$roll\_no:/d" "$STUDENT\_FILE"

echo "$roll\_no:${data[1]}:$new\_marks:${data[3]}:${data[4]}" >> "$STUDENT\_FILE"

echo "Marks assigned successfully!"

else

echo "Student not found!"

fi

sleep 1

}

calculate\_grades\_cgpa() {

clear

echo "===== CALCULATE GRADES & CGPA ====="

temp\_file=$(mktemp)

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

if [ "$marks" -ge 95 ]; then

grade="A+"

grade\_points=4.0

elif [ "$marks" -ge 90 ]; then

grade="A"

grade\_points=4.0

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

grade="A-"

grade\_points=3.7

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

grade="B+"

grade\_points=3.3

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

grade="B"

grade\_points=3.0

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

grade="B-"

grade\_points=2.7

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

grade="C+"

grade\_points=2.3

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

grade="C"

grade\_points=2.0

elif [ "$marks" -ge 57 ]; then

grade="C-"

grade\_points=1.7

elif [ "$marks" -ge 54 ]; then

grade="D+"

grade\_points=1.3

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

grade="D"

grade\_points=1.0

else

grade="F"

grade\_points=0.0

fi

cgpa=$(printf "%.2f" "$grade\_points")

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

done < "$STUDENT\_FILE"

mv "$temp\_file" "$STUDENT\_FILE"

echo "Grades and CGPA calculated for all students!"

sleep 1

}

view\_all\_students() {

clear

echo "===== ALL STUDENTS ====="

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

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

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

printf "%-8s| %-20s| %-6s| %-6s| %-5s\n" "$roll\_no" "$name" "$marks" "$grade" "$cgpa"

done < "$STUDENT\_FILE"

read -p "Press Enter to continue..."

}

view\_passed\_students() {

clear

echo "===== PASSED STUDENTS (CGPA >= 2.0) ====="

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

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

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

if (( $(echo "$cgpa >= 2.0" | bc -l) )); then

printf "%-8s| %-20s| %-6s| %-6s| %-5s\n" "$roll\_no" "$name" "$marks" "$grade" "$cgpa"

fi

done < "$STUDENT\_FILE"

read -p "Press Enter to continue..."

}

view\_failed\_students() {

clear

echo "===== FAILED STUDENTS (CGPA < 2.0) ====="

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

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

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

if (( $(echo "$cgpa < 2.0" | bc -l) )); then

printf "%-8s| %-20s| %-6s| %-6s| %-5s\n" "$roll\_no" "$name" "$marks" "$grade" "$cgpa"

fi

done < "$STUDENT\_FILE"

read -p "Press Enter to continue..."

}

sort\_students\_ascending() {

clear

echo "===== STUDENTS SORTED BY CGPA (ASCENDING) ====="

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

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

sort -t: -k5,5n "$STUDENT\_FILE" | while IFS=: read -r roll\_no name marks grade cgpa; do

printf "%-8s| %-20s| %-6s| %-6s| %-5s\n" "$roll\_no" "$name" "$marks" "$grade" "$cgpa"

done

read -p "Press Enter to continue..."

}

sort\_students\_descending() {

clear

echo "===== STUDENTS SORTED BY CGPA (DESCENDING) ====="

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

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

sort -t: -k5,5nr "$STUDENT\_FILE" | while IFS=: read -r roll\_no name marks grade cgpa; do

printf "%-8s| %-20s| %-6s| %-6s| %-5s\n" "$roll\_no" "$name" "$marks" "$grade" "$cgpa"

done

read -p "Press Enter to continue..."

}

student\_login() {

clear

echo "===== STUDENT LOGIN ====="

view\_all\_students

echo

read -p "Enter your Roll No: " roll\_no

if grep -q "^$roll\_no:" "$STUDENT\_FILE"; then

echo "student" > "$LOGIN\_FILE"

echo "$roll\_no" >> "$LOGIN\_FILE"

echo "Login successful!"

sleep 1

student\_menu

else

echo "Roll No not found!"

sleep 1

main\_menu

fi

}

student\_menu() {

while true; do

clear

roll\_no=$(sed -n '2p' "$LOGIN\_FILE")

student\_data=$(grep "^$roll\_no:" "$STUDENT\_FILE")

IFS=':' read -ra data <<< "$student\_data"

echo "===== STUDENT MENU ====="

echo "Logged in as: ${data[1]} (Roll No: $roll\_no)"

echo "1. View My Grades"

echo "2. View My CGPA"

echo "3. Logout"

echo "4. Exit System"

read -p "Enter your choice: " choice

case $choice in

1) view\_my\_grades ;;

2) view\_my\_cgpa ;;

3) logout ;;

4) exit\_system ;;

\*) echo "Invalid choice!"; sleep 1 ;;

esac

done

}

view\_my\_grades() {

clear

roll\_no=$(sed -n '2p' "$LOGIN\_FILE")

student\_data=$(grep "^$roll\_no:" "$STUDENT\_FILE")

IFS=':' read -ra data <<< "$student\_data"

echo "===== MY GRADES ====="

echo "Name: ${data[1]}"

echo "Roll No: $roll\_no"

echo "Marks: ${data[2]}"

echo "Grade: ${data[3]}"

read -p "Press Enter to continue..."

}

view\_my\_cgpa() {

clear

roll\_no=$(sed -n '2p' "$LOGIN\_FILE")

student\_data=$(grep "^$roll\_no:" "$STUDENT\_FILE")

IFS=':' read -ra data <<< "$student\_data"

echo "===== MY CGPA ====="

echo "Name: ${data[1]}"

echo "Roll No: $roll\_no"

echo "CGPA: ${data[4]}"

read -p "Press Enter to continue..."

}

logout() {

> "$LOGIN\_FILE"

echo "Logged out successfully!"

sleep 1

main\_menu

}

exit\_system() {

echo "Exiting system..."

sleep 1

clear

exit 0

}

main\_menu() {

while true; do

clear

echo "===== STUDENT MANAGEMENT SYSTEM ====="

echo "1. Teacher Login"

echo "2. Student Login"

echo "3. Exit"

read -p "Enter your choice: " choice

case $choice in

1) teacher\_login ;;

2) student\_login ;;

3) exit\_system ;;

\*) echo "Invalid choice!"; sleep 1 ;;

esac

done

}

initialize\_files

main\_menu













