

OPERATING SYSTEMS LAB

FINAL PROJECT - SMS



MUHAMMAD RAMEEZ

23F-0636

# SOURCE CODE

#!/bin/bash

# ---------------------------------------------

# Student Management System (SMS)

# Developed in Bash for a single teacher to manage up to 20 students.

# Features:

# - Secure login system for Teacher and Students

# - CRUD operations on student records (Add, View, Update, Delete)

# - Automatic Grade and CGPA Calculation

# - Students can log in to view their grades and CGPA

# - Data persistence using "students.txt"

# ---------------------------------------------

# ============================

# Constants & Global Variables

# ============================

STUDENT\_FILE="students.txt" # File to store student records

MAX\_STUDENTS=20 # Maximum students allowed

PASS\_CGPA=2.0 # Minimum CGPA required to pass

# ============================

# Load Students Function

# Ensures that the student records file exists, creating it if necessary.

# ============================

load\_students() {

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

touch $STUDENT\_FILE

fi

}

# ============================

# Calculate Grade Function

# Determines the grade based on marks, following FAST grading criteria.

# ============================

calculate\_grade() {

local marks=$1

if (( marks >= 90 )); then echo "A+";

elif (( marks >= 85 )); then echo "A";

elif (( marks >= 80 )); then echo "A-";

elif (( marks >= 75 )); then echo "B+";

elif (( marks >= 70 )); then echo "B";

elif (( marks >= 65 )); then echo "B-";

elif (( marks >= 60 )); then echo "C+";

elif (( marks >= 55 )); then echo "C";

elif (( marks >= 50 )); then echo "D";

else echo "F"; fi

}

# ============================

# Convert Grade to GPA Function

# Converts letter grades into corresponding GPA values.

# ============================

grade\_to\_gpa() {

case $1 in

"A+") echo "4.0";;

"A") echo "4.0";;

"A-") echo "3.7";;

"B+") echo "3.3";;

"B") echo "3.0";;

"B-") echo "2.7";;

"C+") echo "2.3";;

"C") echo "2.0";;

"D") echo "1.0";;

"F") echo "0.0";;

\*) echo "0.0";;

esac

}

# ============================

# Add Student Function

# Allows the teacher to add a new student with marks, grade, and CGPA.

# ============================

add\_student() {

local count=$(wc -l < $STUDENT\_FILE)

if (( count >= MAX\_STUDENTS )); then

echo "Maximum student limit reached!"

return

fi

read -p "Enter Roll Number: " roll

grep -q "^$roll," $STUDENT\_FILE && { echo "Student already exists!"; return; }

read -p "Enter Name: " name

read -p "Enter Marks: " marks

grade=$(calculate\_grade $marks)

gpa=$(grade\_to\_gpa $grade)

echo "$roll,$name,$marks,$grade,$gpa" >> $STUDENT\_FILE

echo "Student added successfully."

}

# ============================

# View All Students Function

# Displays all student records in a formatted table.

# ============================

view\_all\_students() {

echo -e "Roll\tName\tMarks\tGrade\tCGPA"

column -t -s, $STUDENT\_FILE

}

# ============================

# View Student by Roll Number Function

# Allows searching for a specific student's details using Roll Number.

# ============================

view\_student() {

read -p "Enter Roll Number: " roll

grep "^$roll," $STUDENT\_FILE || echo "Student not found!"

}

# ============================

# Delete Student Function

# Removes a student record based on Roll Number.

# ============================

delete\_student() {

read -p "Enter Roll Number to delete: " roll

grep -v "^$roll," $STUDENT\_FILE > temp && mv temp $STUDENT\_FILE

echo "Student deleted (if existed)."

}

# ============================

# Update Marks Function

# Updates a student's marks and recalculates the grade and CGPA.

# ============================

update\_marks() {

read -p "Enter Roll Number: " roll

read -p "Enter new marks: " marks

grade=$(calculate\_grade $marks)

gpa=$(grade\_to\_gpa $grade)

awk -F, -v r=$roll -v m=$marks -v g=$grade -v c=$gpa 'BEGIN{OFS=","}

$1==r{$3=m;$4=g;$5=c} 1' $STUDENT\_FILE > temp && mv temp $STUDENT\_FILE

echo "Marks updated."

}

# ============================

# List Students by CGPA Function

# Displays students sorted in ascending or descending order of CGPA.

# ============================

list\_students\_by\_cgpa() {

echo "1) Ascending"

echo "2) Descending"

read -p "Select order: " order

if [[ $order == 1 ]]; then

sort -t, -k5 -n $STUDENT\_FILE | column -t -s,

else

sort -t, -k5 -nr $STUDENT\_FILE | column -t -s,

fi

}

# ============================

# List Passed/Failed Students Function

# Displays students who have passed or failed based on CGPA.

# ============================

list\_pass\_fail() {

echo "1) Passed"

echo "2) Failed"

read -p "Select option: " opt

if [[ $opt == 1 ]]; then

awk -F, -v th=$PASS\_CGPA '$5 >= th' $STUDENT\_FILE | column -t -s,

else

awk -F, -v th=$PASS\_CGPA '$5 < th' $STUDENT\_FILE | column -t -s,

fi

}

# ============================

# Teacher Login Function

# Authenticates the teacher with a predefined password.

# ============================

teacher\_login() {

TEACHER\_PASSWORD="fast123" # Set teacher password

read -s -p "Enter Password: " input\_pass

echo

if [[ "$input\_pass" == "$TEACHER\_PASSWORD" ]]; then

teacher\_menu

else

echo "Incorrect Password!"

fi

}

# ============================

# Student Login Function

# Authenticates students using Roll Number and default password.

# ============================

student\_login() {

read -p "Enter Your Roll Number: " roll

student=$(grep "^$roll," $STUDENT\_FILE)

if [[ -z "$student" ]]; then

echo "Student not found!"

return

fi

last4="${roll: -4}"

expected\_pass="student@$last4"

read -s -p "Enter Password: " input\_pass

echo

if [[ "$input\_pass" == "$expected\_pass" ]]; then

IFS=',' read -r r name marks grade gpa <<< "$student"

echo "Name: $name"

echo "Marks: $marks"

echo "Grade: $grade"

echo "CGPA: $gpa"

else

echo "Incorrect Password!"

fi

}

# ============================

# Teacher Menu Function

# Provides options for managing students.

# ============================

teacher\_menu() {

while true; do

echo -e "\n--- Teacher Menu ---"

echo "1) Add Student"

echo "2) View All Students"

echo "3) View Student"

echo "4) Update Marks"

echo "5) Delete Student"

echo "6) List Students by CGPA"

echo "7) List Passed/Failed Students"

echo "0) Logout"

read -p "Choose: " choice

case $choice in

1) add\_student ;;

2) view\_all\_students ;;

3) view\_student ;;

4) update\_marks ;;

5) delete\_student ;;

6) list\_students\_by\_cgpa ;;

7) list\_pass\_fail ;;

0) break ;;

\*) echo "Invalid option." ;;

esac

done

}

#main\_menu

main\_menu() {

clear

# Color codes

GREEN='\033[0;32m'

BLUE='\033[1;34m'

RED='\033[0;31m'

NC='\033[0m' # No Color

# Print Banner

echo -e "${BLUE}"

echo " \_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_ "

echo " / \_\_\_\_\_) (\_\_\_\_\_\_\_) / \_\_\_\_\_)"

echo " ( (\_\_\_\_ \_ \_ \_ ( (\_\_\_\_ "

echo " \\_\_\_\_ \ | ||\_|| | \\_\_\_\_ \ "

echo " \_\_\_\_\_) ) | | | | \_\_\_\_\_) )"

echo " (\_\_\_\_\_\_/ |\_| |\_| (\_\_\_\_\_\_/ "

echo -e "${GREEN}"

echo " Welcome to the Student Management System"

echo " Built with ❤️ by RAMEEZ"

echo -e "${NC}"

sleep 1

load\_students

while true; do

echo -e "\n🔐 ${BLUE}Main Menu:${NC}"

echo "1) 👨‍🏫 Teacher Login"

echo "2) 👨‍🎓 Student Login"

echo "0) 🚪 Exit"

read -p "Choose: " opt

case $opt in

1) teacher\_login ;;

2) student\_login ;;

0) echo -e "${GREEN}Goodbye!${NC}"; break ;;

\*) echo -e "${RED}Invalid option!${NC}" ;;

esac

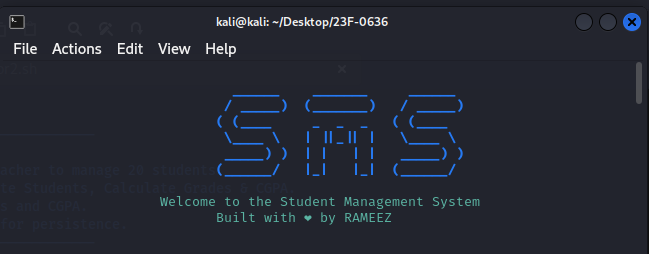
done

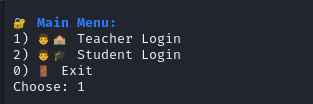
}

# Run the main menu

main\_menu

# SCREENSHOTS



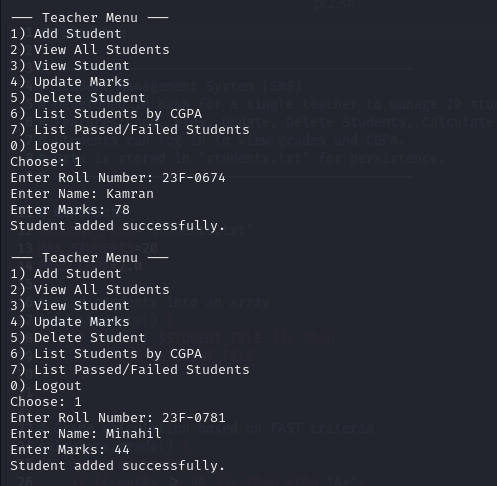


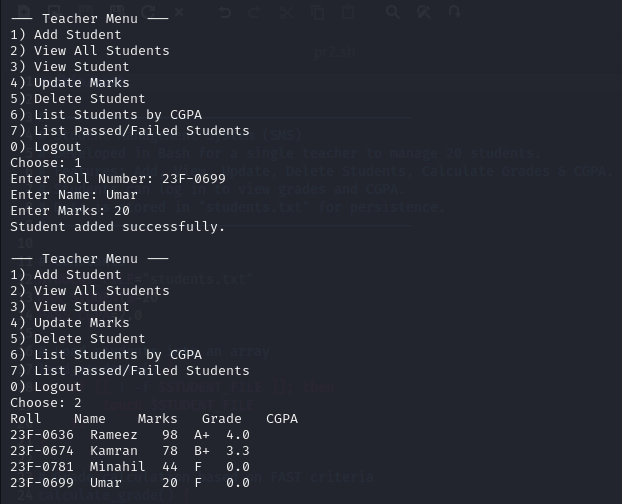
A screenshot of a computer

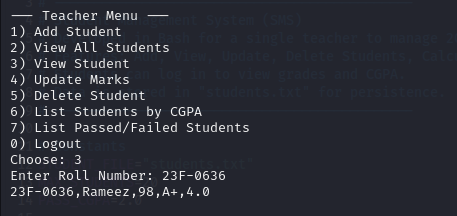
Description automatically generated

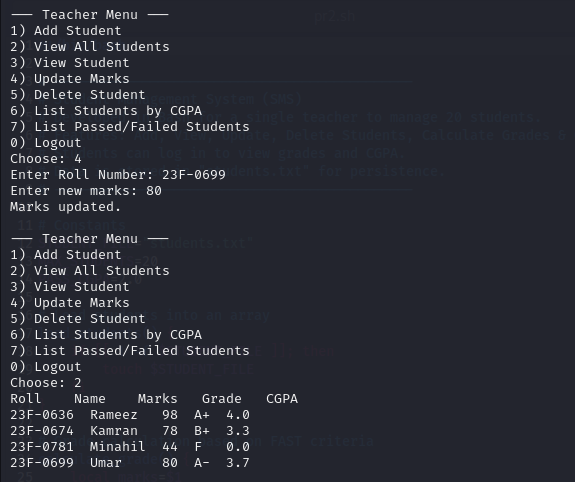
THE DEFAULT PASSWORD FOR TEACHER IS “fast123”

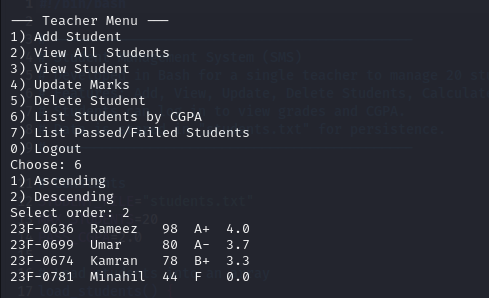
THIS CAN BE CHANGED FROM THE SCRIPT

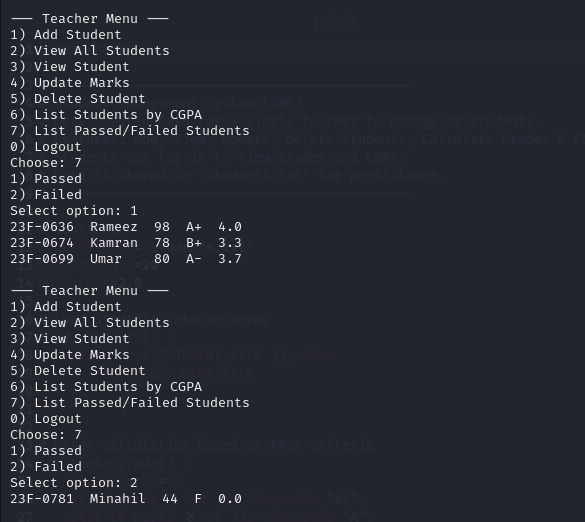


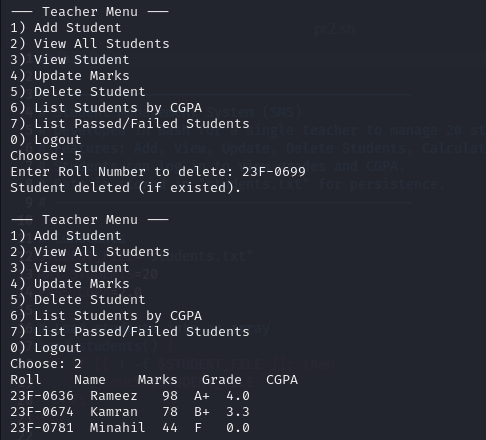


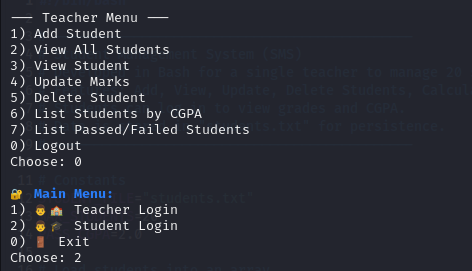


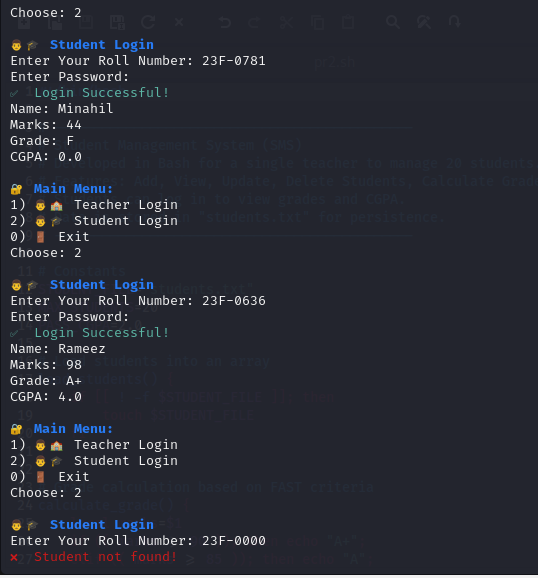










THE PASSWORD FOR EVERY STUDENT IS DIFFERENT PASSWROD FOR EACH STUDENT IS “student@(last 4 digits of roll no)”

