**OPERATING SYSTEMS**

**LAB PROJECT**

**Group Members:** 23F-0738 & 21L-5097

**CODE:**

#!/bin/bash

# file to store student data

data\_file="students.txt"

# some constants for limits and gpa stuff

max\_students=20

credit\_hours=3

total\_credits=9

passing\_cgpa=2.0

# arrays to hold student data and login users

declare -A students

declare -A users

# logins credentials

users[teacher]="T1234"

users[student]="S123"

# function to check negative  
is\_positive\_integer() {

[[ "$1" =~ ^[0-9]+$ ]]

}

# load data from file into memory

load\_data() {

if [[ -f $data\_file ]]; then

while IFS="|" read -r roll name os db sda grade\_os grade\_db grade\_sda gpa; do

students[$roll]="$name|$os|$db|$sda|$grade\_os|$grade\_db|$grade\_sda|$gpa"

done < "$data\_file"

fi

}

# save everything back to the file

save\_data() {

> $data\_file # clear the file first

for roll in "${!students[@]}"; do

echo "$roll|${students[$roll]}" >> $data\_file

done

}

# login screen

login() {

read -p "enter username: " username

read -s -p "enter password: " password

echo ""

if [[ "${users[$username]}" == "$password" ]]; then

echo "login successful!"

if [[ "$username" == "teacher" ]]; then

teacher\_menu

else

student\_menu

fi

else

echo "invalid credentials!"

login # try again

fi

}

# add a student

add\_student() {

if [[ ${#students[@]} -ge $max\_students ]]; then

echo "sorry, max students reached!"

return

fi

read -p "enter roll number: " roll

if ! is\_positive\_integer "$roll"; then

echo "invalid roll number. must be a non-negative integer."

return

fi

if [[ -n "${students[$roll]}" ]]; then

echo "student already exists."

return

fi

read -p "enter name: " name

read -p "enter marks for os: " os

read -p "enter marks for db: " db

read -p "enter marks for sda: " sda

if ! is\_positive\_integer "$os" || ! is\_positive\_integer "$db" || ! is\_positive\_integer "$sda"; then

echo "marks must be non-negative integers."

return

fi

# calculate grades

grade\_os=$(calculate\_grade "$os")

grade\_db=$(calculate\_grade "$db")

grade\_sda=$(calculate\_grade "$sda")

# calculate gpa

gpa=$(calculate\_gpa "$grade\_os" "$grade\_db" "$grade\_sda")

# save student record

students[$roll]="$name|$os|$db|$sda|$grade\_os|$grade\_db|$grade\_sda|$gpa"

echo "student added!"

}

# delete a student

delete\_student() {

read -p "enter roll number to delete: " roll

unset students[$roll]

echo "student deleted."

}

# update student marks

update\_student() {

read -p "enter roll number: " roll

read -p "enter new marks for os: " os

read -p "enter new marks for db: " db

read -p "enter new marks for sda: " sda

if ! is\_positive\_integer "$os" || ! is\_positive\_integer "$db" || ! is\_positive\_integer "$sda"; then

echo "marks must be non-negative integers."

return

fi

IFS='|' read -r name \_ <<< "${students[$roll]}"

grade\_os=$(calculate\_grade "$os")

grade\_db=$(calculate\_grade "$db")

grade\_sda=$(calculate\_grade "$sda")

gpa=$(calculate\_gpa "$grade\_os" "$grade\_db" "$grade\_sda")

students[$roll]="$name|$os|$db|$sda|$grade\_os|$grade\_db|$grade\_sda|$gpa"

echo "record updated!"

}

# grade from marks

calculate\_grade() {

marks=$1

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

elif (( marks >= 80 )); then echo "B"

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

elif (( marks >= 60 )); then echo "D"

else echo "F"

fi

}

# convert grade to points

grade\_to\_point() {

grade=$1

case $grade in

A) echo "4.0" ;;

B) echo "3.5" ;;

C) echo "3.0" ;;

D) echo "2.0" ;;

F) echo "1.0" ;;

esac

}

# gpa calculator

calculate\_gpa() {

g1=$(grade\_to\_point $1)

g2=$(grade\_to\_point $2)

g3=$(grade\_to\_point $3)

total=$(echo "scale=2; ($g1 + $g2 + $g3) \* $credit\_hours / $total\_credits" | bc)

echo "$total"

}

# view student by roll

view\_student() {

read -p "enter roll number: " roll

if [[ -z "${students[$roll]}" ]]; then

echo "no student found."

return

fi

IFS='|' read -r name os db sda g\_os g\_db g\_sda gpa <<< "${students[$roll]}"

echo "name: $name"

echo "os: $os, grade: $g\_os"

echo "db: $db, grade: $g\_db"

echo "sda: $sda, grade: $g\_sda"

echo "gpa: $gpa"

}

# just show gpa

view\_cgpa() {

read -p "enter roll number: " roll

if [[ -z "${students[$roll]}" ]]; then

echo "no student found."

return

fi

IFS='|' read -r \_ \_ \_ \_ \_ \_ \_ gpa <<< "${students[$roll]}"

echo "gpa: $gpa"

}

# display all students

display\_students() {

echo "roll | name | os | db | sda | grade os | grade db | grade sda | gpa"

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

for roll in "${!students[@]}"; do

IFS='|' read -r name os db sda g\_os g\_db g\_sda gpa <<< "${students[$roll]}"

echo "$roll | $name | $os | $db | $sda | $g\_os | $g\_db | $g\_sda | $gpa"

done | column -t -s '|'

}

# sort students based on gpa

sort\_students() {

read -p "sort by gpa (a for ascending, d for descending): " order

if [[ "$order" == "a" || "$order" == "A" ]]; then

sort\_flag="-k9,9"

else

sort\_flag="-k9,9 -r"

fi

echo "roll | name | os | db | sda | grade os | grade db | grade sda | gpa"

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

for roll in "${!students[@]}"; do

echo "$roll|${students[$roll]}"

done | sort -t '|' $sort\_flag | column -t -s '|'

}

# show who passed and who didn’t

list\_passed\_failed() {

echo "passed students:"

for roll in "${!students[@]}"; do

IFS='|' read -r name os db sda g\_os g\_db g\_sda gpa <<< "${students[$roll]}"

if (( $(echo "$gpa >= $passing\_cgpa" | bc -l) )); then

echo "$roll | $name | gpa: $gpa"

fi

done | column -t -s '|'

echo "failed students:"

for roll in "${!students[@]}"; do

IFS='|' read -r name os db sda g\_os g\_db g\_sda gpa <<< "${students[$roll]}"

if (( $(echo "$gpa < $passing\_cgpa" | bc -l) )); then

echo "$roll | $name | gpa: $gpa"

fi

done | column -t -s '|'

}

# menu for teacher login

teacher\_menu() {

while true; do

echo "===== teacher menu ====="

echo "1. add student"

echo "2. delete student"

echo "3. update student marks"

echo "4. view student details"

echo "5. display all students"

echo "6. sort students by gpa"

echo "7. list passed & failed"

echo "8. save data"

echo "9. load data"

echo "0. logout"

read -p "choose an option: " choice

case $choice in

1) add\_student ;;

2) delete\_student ;;

3) update\_student ;;

4) view\_student ;;

5) display\_students ;;

6) sort\_students ;;

7) list\_passed\_failed ;;

8) save\_data ;;

9) load\_data ;;

0) echo "logging out..."; return ;;

\*) echo "invalid option" ;;

esac

done

}

# menu for student login

student\_menu() {

while true; do

echo "===== student menu ====="

echo "1. view grades"

echo "2. view gpa"

echo "0. logout"

read -p "choose an option: " choice

case $choice in

1) view\_student ;;

2) view\_cgpa ;;

0) echo "logging out..."; return ;;

\*) echo "invalid option" ;;

esac

done

}

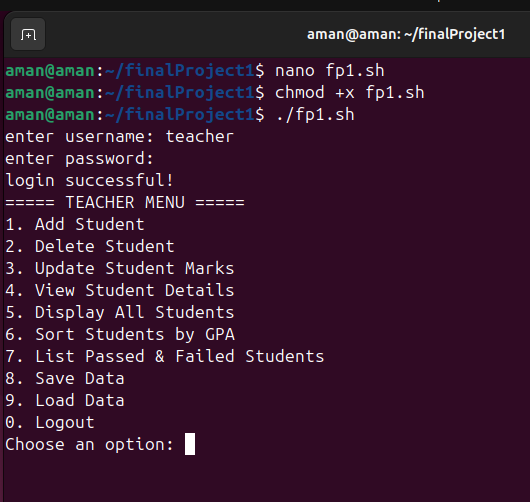
# load existing data and start login

load\_data

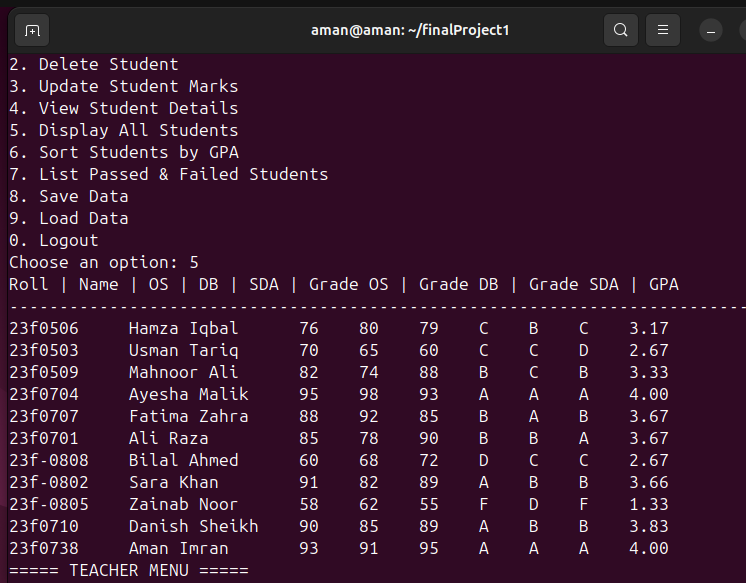
login

**SCREENSHOTS:**

**For Teachers:**

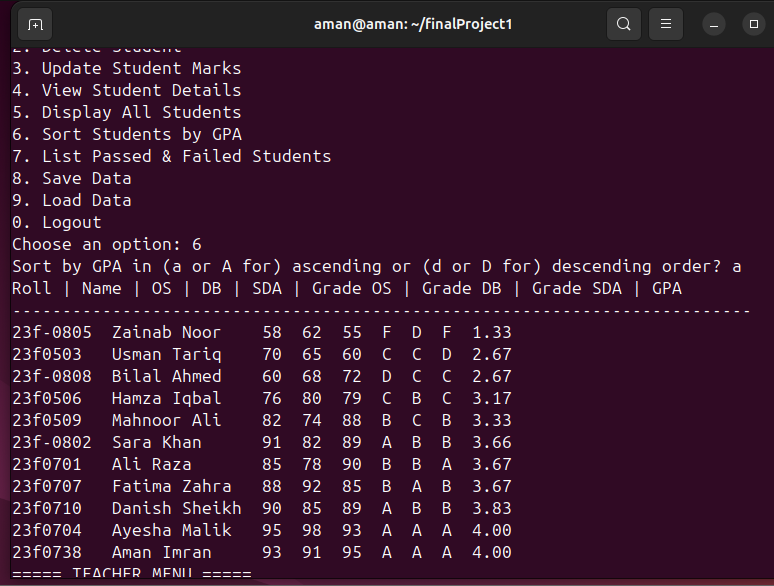
**Login:**

**Display students:**

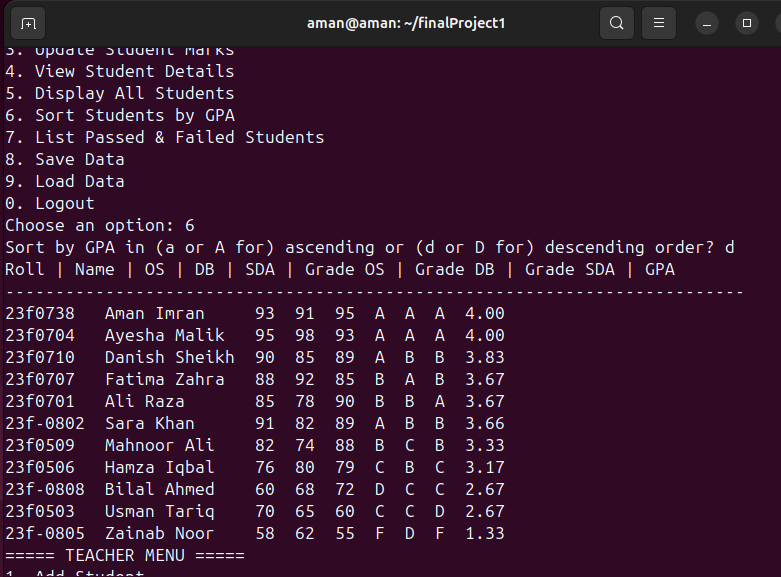


**Sorting students:**

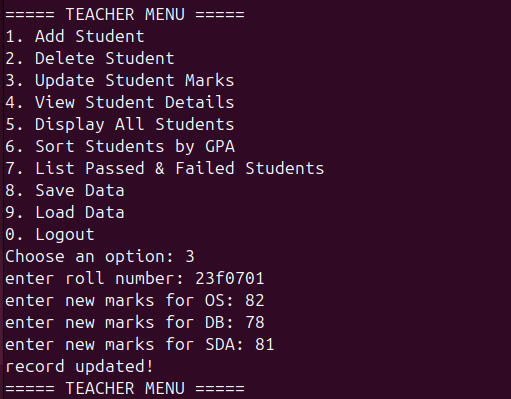
**ascending:**



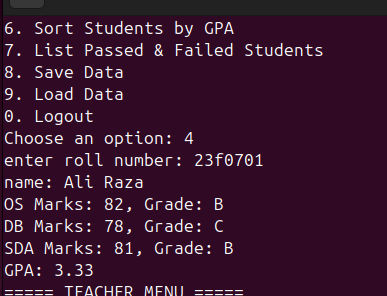
**Descending:**



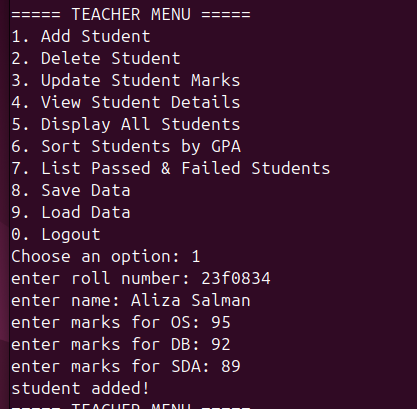
**Update marks:**



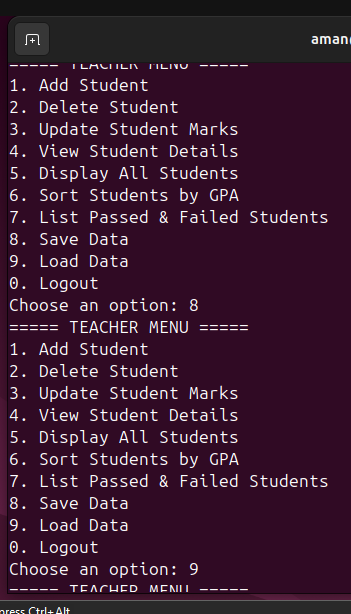
**updated marks shown:**



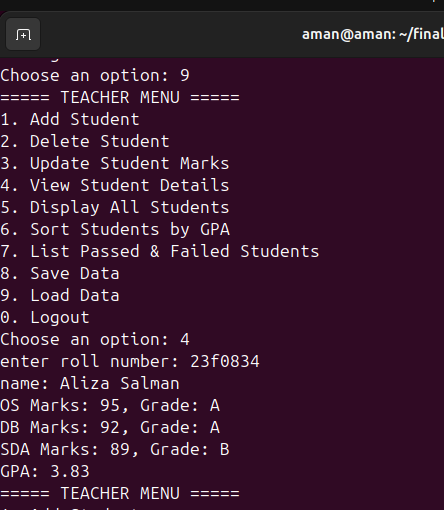
**Add student:**



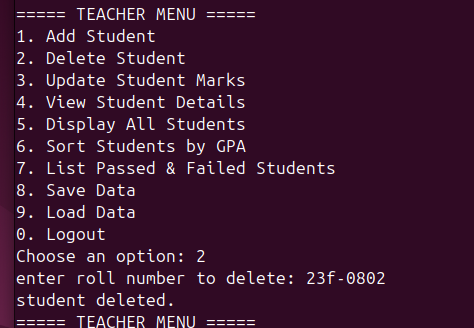
**Save and Load data:**

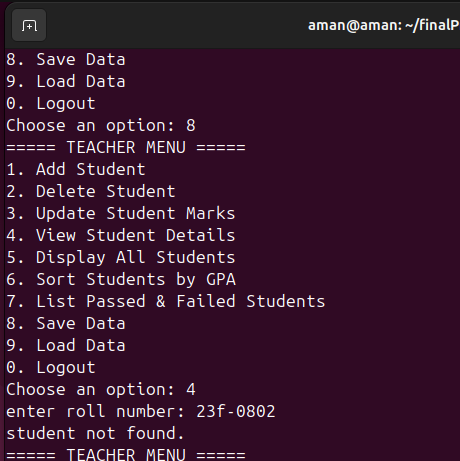


**View student:**

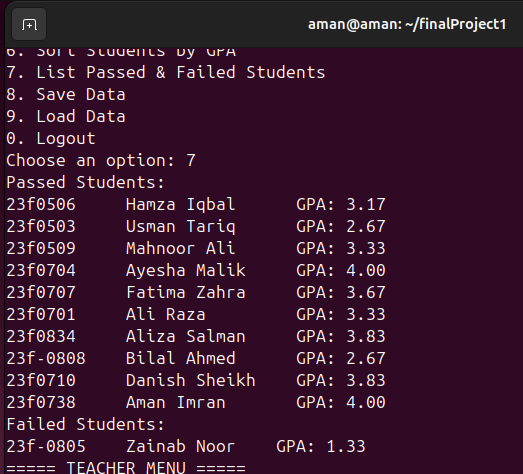


**Delete student:**

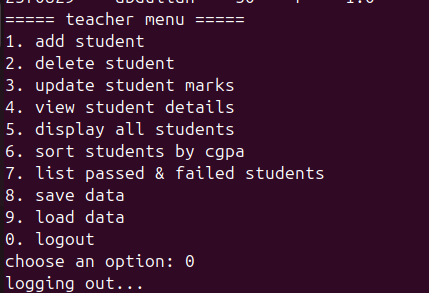


**After deletion:**

**Fail and Passed students:**

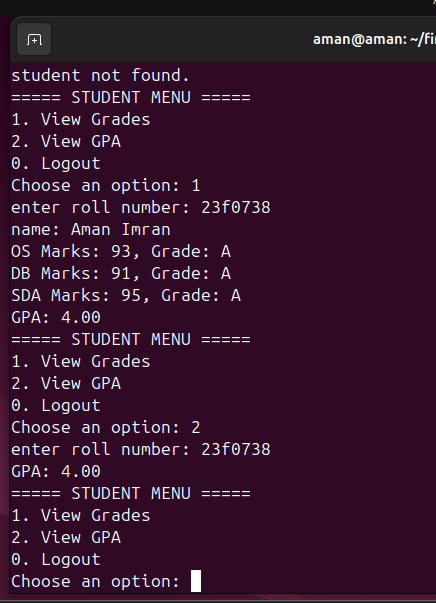


**Log out:**



**For Students:**

**View grade and cgpa:**



**TXT FILE DATA:**

**Students.txt:**

23f0506|Hamza Iqbal|76|80|79|C|B|C|3.17

23f0503|Usman Tariq|70|65|60|C|C|D|2.67

23f0509|Mahnoor Ali|82|74|88|B|C|B|3.33

23f0704|Ayesha Malik|95|98|93|A|A|A|4.00

23f0707|Fatima Zahra|88|92|85|B|A|B|3.67

23f0701|Ali Raza|82|78|81|B|C|B|3.33

23f0834|Aliza Salman|95|92|89|A|A|B|3.83

23f0808|Bilal Ahmed|60|68|72|D|C|C|2.67

23f0805|Zainab Noor|58|62|55|F|D|F|1.33

23f0710|Danish Sheikh|90|85|89|A|B|B|3.83

23f0738|Aman Imran|93|91|95|A|A|A|4.00