

**Lab Report: Student Portal Using Shell Script**

Course Code: CIS232L

Course Title: Operating Systems Lab

**Submitted to**

Ms. Sonia Nasrin

Lecturer

Department of CIS

Daffodil International University

**Submitted by**

Ishita Islam

222-16-658

Department of CIS

Daffodil International University

Date of Submission 04.11.2023

# Introduction

The Student Management System is a software application designed to streamline the administrative and academic tasks of educational institutions. It serves as a comprehensive tool for administrators, teachers, and students to manage student information, course assignments, grades, and other essential functions related to academic operations. This lab report provides an overview of the Student Management System, outlining its motivation, objectives, tools used, methodology, and concluding remarks.

# Motivation

The motivation behind developing the Student Management System is to address the challenges faced by educational institutions in managing student data and academic operations efficiently. Manual record-keeping and communication can be error-prone, time-consuming, and lead to inefficiencies. The system aims to automate and simplify administrative and academic tasks, enhance data accuracy, and provide a more convenient user experience for administrators, teachers, and students.

# Objective

The primary objectives of the Student Management System project are as follows:

1. Student Information Management: To create a database for storing student records, including personal information, academic history, and enrollment details.

2. Teacher Information Management: To maintain a database of teacher records, including their personal details and assigned courses.

3. Course Management: To facilitate the creation and management of courses, including assigning teachers to courses and enrolling students.

4. Marks Management: To allow teachers to update and manage students' marks for specific courses.

5. User-Friendly Interfaces: To provide intuitive interfaces for administrators, teachers, and students, making it easy to access and manage their respective information.

6. Security: To ensure data security and access control through authentication mechanisms and password protection.

# Tools

The development of the Student Management System involves the use of the following tools and technologies:

1. Bash Scripting: To create a command-line interface for the system, handling user interactions and database operations.

2. CSV Files: To store data related to students, teachers, courses, semesters, and marks.

3. PlantUML: To generate UML diagrams representing the project's structure and relationships.

4. Text Editors: To write and edit Bash scripts and documentation.

# Methodology

The development of the Student Management System follows the following methodology:

1. Data Management: CSV files are used to store data for students, teachers, courses, semesters, and marks. Bash scripts are created to read and manipulate these files for data management.

2. User Authentication: The system ensures user authentication for administrators, teachers, and students. Passwords are securely stored and verified.

3. Menu-Driven Interface: The system employs a menu-driven interface to provide a user-friendly experience, with distinct options for each user type.

4. Data Validation: Data validation is implemented to prevent inconsistencies and errors, ensuring the integrity of student, teacher, course, and marks data.

5. Reports: UML diagrams are generated using PlantUML to provide visual representations of the project's structure.

6. Testing and Debugging: Extensive testing and debugging are performed to identify and resolve issues, ensuring the system's reliability.

# Implementation

Source Code:

#!/bin/bash

STUDENT\_DATA="students.csv"

TEACHER\_DATA="teachers.csv"

COURSE\_DATA="courses.csv"

SEMESTER\_DATA="semesters.csv"

MARKS\_DATA="marks.csv"

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

    touch $STUDENT\_DATA

fi

if [ ! -f $TEACHER\_DATA ]; then

    touch $TEACHER\_DATA

fi

if [ ! -f $COURSE\_DATA ]; then

    touch $COURSE\_DATA

fi

if [ ! -f $SEMESTER\_DATA ]; then

    touch $SEMESTER\_DATA

fi

if [ ! -f $MARKS\_DATA ]; then

    touch $MARKS\_DATA

fi

add\_student() {

    echo "Enter Student Name: "

    read name

    echo "Enter Student ID: "

    read id

    echo "Enter Student Password: "

    read password

    echo "$id,$name,$password" >> $STUDENT\_DATA

    echo "Student added successfully."

}

# Function to add a new teacher

add\_teacher() {

    echo "Enter Teacher Name: "

    read name

    echo "Enter Teacher ID: "

    read id

    echo "Enter Teacher Password: "

    read password

    echo "$id,$name,$password" >> $TEACHER\_DATA

    echo "Teacher added successfully."

}

create\_course() {

    echo "Enter Course Name: "

    read name

    echo "Enter Course Code: "

    read code

    echo "$code,$name" >> $COURSE\_DATA

    echo "Course created successfully."

}

assign\_teacher() {

    echo "Enter Teacher ID: "

    read teacher\_id

    echo "Enter Course Code: "

    read course\_code

    if grep -q "$teacher\_id" $TEACHER\_DATA && grep -q "$course\_code" $COURSE\_DATA; then

        echo "$teacher\_id,$course\_code" >> $SEMESTER\_DATA

        echo "Teacher assigned to course successfully."

    else

        echo "Teacher or course not found."

    fi

}

enroll\_student() {

    echo "Enter Student ID: "

    read student\_id

    echo "Enter Course Code: "

    read course\_code

    if grep -q "$student\_id" $STUDENT\_DATA && grep -q "$course\_code" $COURSE\_DATA; then

        echo "$student\_id,$course\_code" >> $SEMESTER\_DATA

        echo "Student enrolled in the course successfully."

    else

        echo "Student or course not found."

    fi

}

delete\_student() {

    echo "Enter Student ID to delete: "

    read student\_id

    grep -v "$student\_id" $STUDENT\_DATA > temp.csv

    mv temp.csv $STUDENT\_DATA

    grep -v "$student\_id" $SEMESTER\_DATA > temp.csv

    mv temp.csv $SEMESTER\_DATA

    echo "Student deleted successfully."

}

view\_students() {

    echo "Student ID,Student Name"

    cat $STUDENT\_DATA

}

view\_teachers() {

    echo "Teacher ID,Teacher Name"

    cat $TEACHER\_DATA

}

view\_courses() {

    echo "Course Code,Course Name"

    cat $COURSE\_DATA

}

view\_semester() {

    echo "ID,Student ID,Course Code"

    cat $SEMESTER\_DATA

}

view\_marks() {

    echo "Enter Student ID: "

    read student\_id

    echo "Enter Course Code: "

    read course\_code

    if grep -q "$student\_id" $STUDENT\_DATA && grep -q "$course\_code" $COURSE\_DATA; then

        grep "$student\_id,$course\_code" $MARKS\_DATA

    else

        echo "Student or course not found."

    fi

}

while true; do

    echo "Student Protal"

    echo "1. Admin"

    echo "2. Teacher"

    echo "3. Student"

    echo "4. Exit"

    echo "Select a user type: "

    read user\_type

    case $user\_type in

        1)

            echo "Enter Admin Password: "

            read admin\_password

            if [ "$admin\_password" = "cis12345" ]; then

                while true; do

                    echo "Admin Menu"

                    echo "1. Add Student"

                    echo "2. Add Teacher"

                    echo "3. Delete Student"

                    echo "4. Create Course"

                    echo "5. Assign Teacher to Course"

                    echo "6. Enroll Student in Course"

                    echo "7. View Students"

                    echo "8. View Teachers"

                    echo "9. View Courses"

                    echo "10. View Semester"

                    echo "11. Exit Admin Menu"

                    echo "Select an option: "

                    read admin\_option

                    case $admin\_option in

                        1) add\_student ;;

                        2) add\_teacher ;;

                        3) delete\_student ;;

                        4) create\_course ;;

                        5) assign\_teacher ;;

                        6) enroll\_student ;;

                        7) view\_students ;;

                        8) view\_teachers ;;

                        9) view\_courses ;;

                        10) view\_semester ;;

                        11) break ;;

                        \*) echo "Invalid option" ;;

                    esac

                done

            else

                echo "Invalid Admin Password"

            fi

            ;;

        2)

            echo "Enter Teacher ID: "

            read teacher\_id

            echo "Enter Teacher Password: "

            read teacher\_password

            if grep -q "$teacher\_id,$teacher\_password" $TEACHER\_DATA; then

                while true; do

                    echo "Teacher Menu"

                    echo "1. Update Marks"

                    echo "2. View Marks"

                    echo "3. Exit Teacher Menu"

                    echo "Select an option: "

                    read teacher\_option

                    case $teacher\_option in

                        1)

                            echo "Enter Student ID: "

                            read student\_id

                            echo "Enter Course Code: "

                            read course\_code

                            echo "Enter Marks: "

                            read marks

                            if grep -q "$student\_id,$course\_code" $SEMESTER\_DATA; then

                                echo "$student\_id,$course\_code,$marks" >> $MARKS\_DATA

                                echo "Marks updated successfully."

                            else

                                echo "Student not enrolled in the course."

                            fi

                            ;;

                        2) view\_marks ;;

                        3) break ;;

                        \*) echo "Invalid option" ;;

                    esac

                done

            else

                echo "Invalid Teacher ID or Password"

            fi

            ;;

        3)

            echo "Enter Student ID: "

            read student\_id

            echo "Enter Student Password: "

            read student\_password

            if grep -q "$student\_id,$student\_password" $STUDENT\_DATA; then

                while true; do

                    echo "Student Menu"

                    echo "1. View Marks"

                    echo "2. Exit Student Menu"

                    echo "Select an option: "

                    read student\_option

                    case $student\_option in

                        1) view\_marks ;;

                        2) break ;;

                        \*) echo "Invalid option" ;;

                    esac

                done

            else

                echo "Invalid Student ID or Password"

            fi

            ;;

        4)

            exit

            ;;

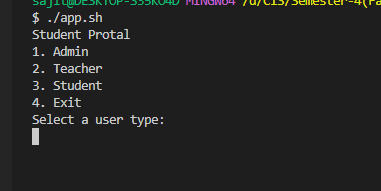
        \*) echo "Invalid user type" ;;

    esac

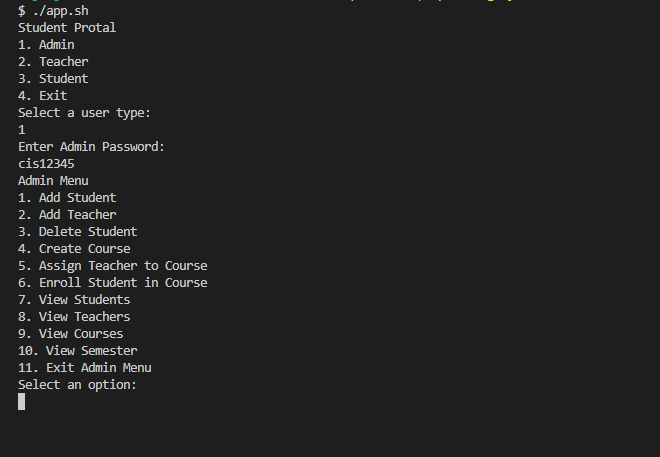
done

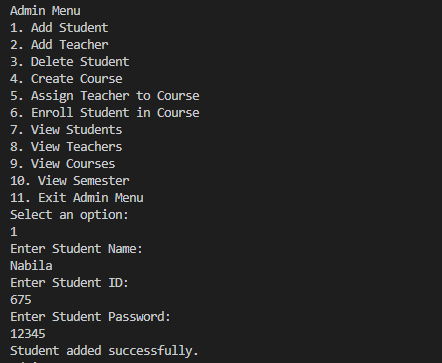
Output:

Main Menu:

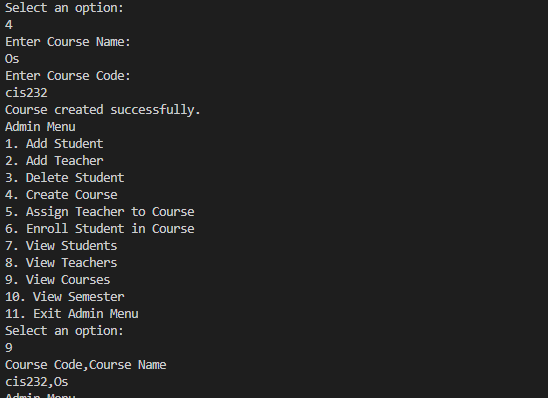


Admin Login:

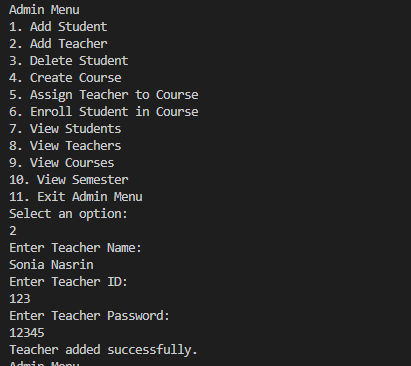


Adding Student:  


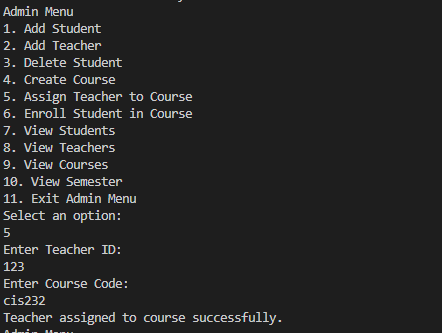
Adding Course:

****

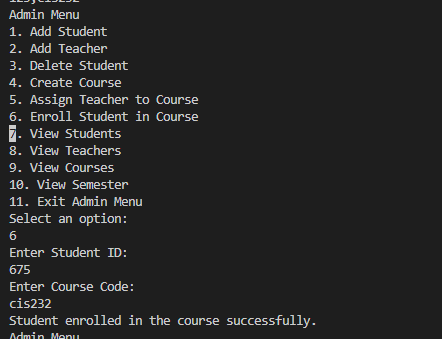
Adding Teacher:

****

Assigning Teacher:

****

Enrolling Student:

****

# Conclusion

The Student Management System is a significant step toward enhancing the efficiency and accuracy of academic and administrative operations in educational institutions. By providing a structured and automated approach to managing student data, course assignments, and grades, the system offers numerous benefits to users. Its user-friendly interfaces, data security measures, and data validation mechanisms contribute to a more streamlined educational environment.

In conclusion, the Student Management System project represents a practical application of technology to address real-world challenges in education and offers a solid foundation for future advancements in educational management.