National University
of computer and emerging sciences

Final Project Report

Programming Fundamentals

Semester Project

Arqam Hassan (21K-3826)

Hassan Amir (21K-3802)

Ali Zain (21K-4653)

Section 1M

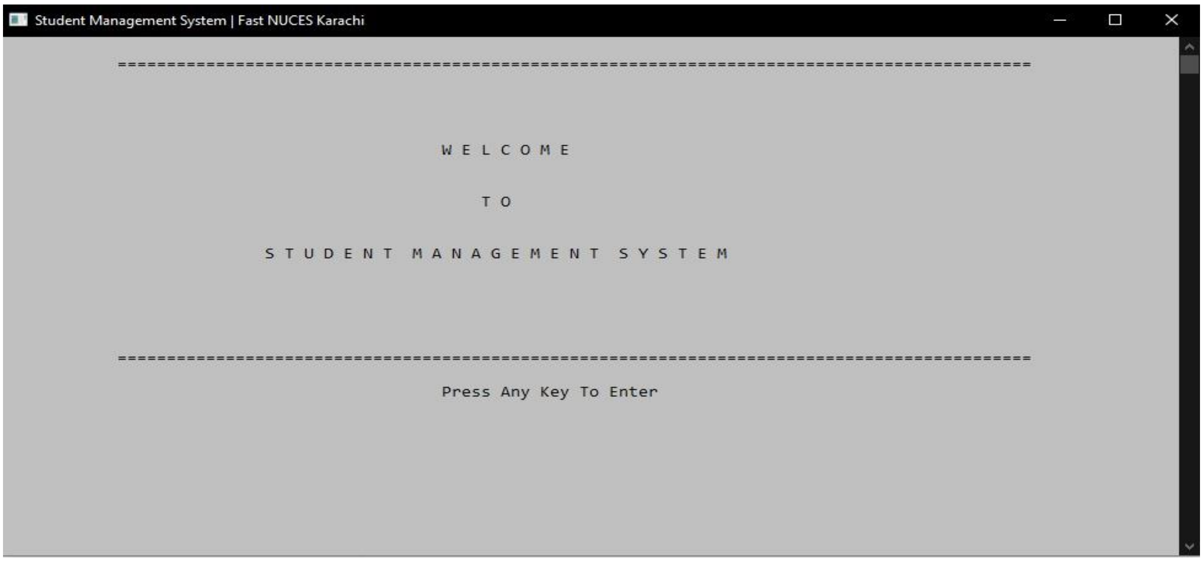
Submitted to:

Miss Atiya Jokhio

Department of Computer Science BS(CS)

FAST-NUCES Karachi

Student Management System



```
          ============================================================================================


                                         W E L C O M E

                                             T O

                              S T U D E N T   M A N A G E M E N T   S Y S T E M



          ============================================================================================
                                         Press Any Key To Enter
```

Introduction

In many institutes around the world, whether it be a prestigious one such as Harvard, or an average school or college. One thing they all have in common is a data management system to store all data related to the students. That is what we have created. A user-friendly system which allows a user to create, read, update and/or delete student data with minimal difficulty. It can be used by both admins and students for their respective purposes.

Background

We went through the provided project list and googled all of them. We watched multiple YouTube videos and reviewed their designs, after which we decided on the student data management system. We chose this as it felt relatively easy to understand and we had a good concept of it. We watched a few videos on the topic, read some articles and took reference from sites such as Flex etc. After which we started working on the project. There were some minor errors and one or two significant errors, but they were resolved in due time and the project making smoothly continued.

Project Specifications

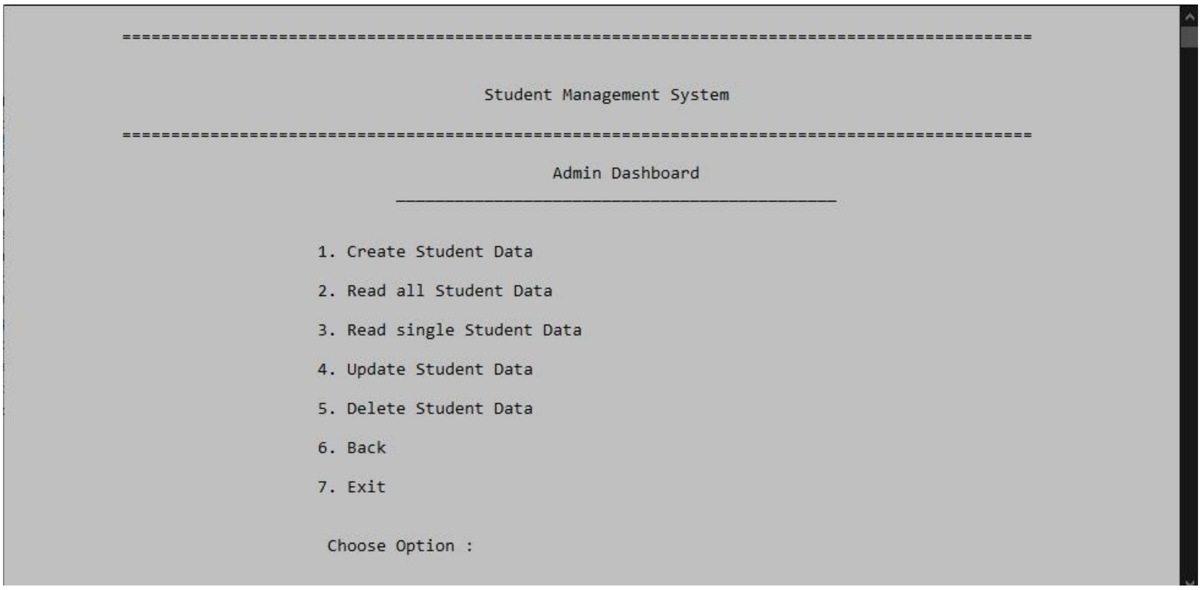
The programming language that was used in coding this project was C programming language. A structure was made that took all values inputted later into their specific locations for further use within the program. The structure is as shown:

```
struct Student {
   char name[50];
   int batch;
   char dep[50];
   char campus[50];
   int roll;
   float gpa[8];
   float cgpa;
} s ;
```

Firstly, the introduction of the program is displayed, after pressing enter, the program will take you to the Admin and Student identification screen. Any one of the two can be chosen after which a password will be requested to input, which will take the user to the respective screen. The commands will be displayed as shown.

-Admin Commands:

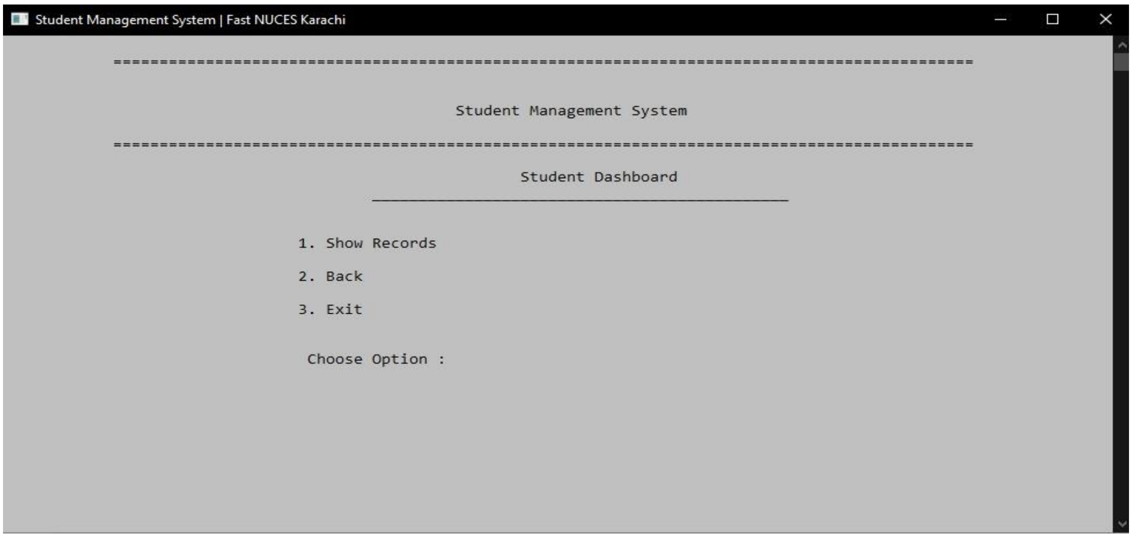
- **1. Create Student Data:** Allows user to enter new student information.
- **2. Read All Student Data:** Allows user to display all entered data.
- **3. Read Single Student Data:** Allows user to display specific student data by inputting roll number.
- **4. Update Student Data:** Allows user to change an already inputted student's data.
- **5. Delete Student Data:** Allows user to delete a specific student's data by inputting roll number.
- **6. Back:** Takes user back to the main screen.
- **7. Exit:** Exits and closes the program.



```
            ============================================================================================

                                          Student Management System

            ============================================================================================

                                                 Admin Dashboard
                                        _____________________________________________


                              1. Create Student Data

                              2. Read all Student Data

                              3. Read single Student Data

                              4. Update Student Data

                              5. Delete Student Data

                              6. Back

                              7. Exit

                               Choose Option :
```

-Student Commands:

- **1. Show Records:** Allows user to display one's student information by inputting roll number.
- **2. Back:** Takes user back to the main screen.
- **3. Exit:** Exits and closes the program.



```
=============================================================================================

                                     Student Management System

=============================================================================================
                                            Student Dashboard
                            _____________________________________________


                    1. Show Records

                    2. Back

                    3. Exit

                     Choose Option :
```

Problem Analysis

The main problems encountered in the making of the project was Filing. As it was a new concept, we had to google and go through many articles and videos to grasp the concept. Keeping track of all the functions was another challenge, there were quite a few minor errors such as missing ampersands, semi-colons, compiler errors etc. One main error we encountered was how to allocate the files and implement pointer filing within each function.

Our code solves the issue of keeping track of the data of multiple students within an institute as it is tiring to find specific student data without a management system and the data can easily get jumbled if not properly kept in order. Our program allows a user to easily create a student's data and at any time display, update or delete that data.

Solution Design

-Project Detail: Our program allows a user to choose an option between Admin and Student. Then asks to input a password respective to the user's choice, after which the user will be taken to the respective dashboard to choose the command which they wish to run. The commands implement basic CRUD Operations. They are as follows:

IF ADMIN: -

{MAIN FUNCTIONS}:

- Create Student Data
- Read Student Data
- Read Single Student Data
- Update Student Data
- Delete Student Data

{SIDE FUNCTIONS}:

- Back
- Exit

IF STUDENT: -

{MAIN FUNCTION}:

- Show Specific Student Data

{SIDE FUNCTIONS}:

- Back
- Exit

-Functionality and Detail: We made multiple functions to implement the crud commands and some other functions to make the program more user friendly.

A total of 8 main functions were made which are:

- **Int_main: -** Primarily consists of a switch case within a while loop to ask user to choose Admin or Student. After choosing one they are taken to the screen respective of their choice via another function implemented in the case.

- **admin_window: -** The function asks user to input a password respective of the choice they made in **Int_main**, after which they are taken to a screen where all available commands are shown. A **switch case** within a **while loop** is used that displays all commands and inputs a specific command from the user after which the user is taken to the chosen command screen due to the functions within **switch case**.

- **student_window: -** This function works similarly to the **admin_window** function with the exception that only one main function is used within the **switch case**.

- **add_std: -** Comprises of a **while loop** which consists of multiple inputs to ask user for student information, a **for loop** to input GPA and calculates CGPA. Filing used to save all the inputted student data.

- **read_std:** - Primarily consists of a **for loop** within a **while loop** which displays all student information that was inputted in the **add_std** function.

- **single_read: -** Primarily consists of a **nested while loop** which displays a specific student's information by input of roll number within first while loop. The inputted roll number is compared to other roll numbers in the structure via an **if** in the **nested while loop**. An **if else** is used to then print the specific student data or a message is shown if given roll number is not found within structure.

- **update_std: -** Asks user to input roll number of student who's data they want to update, which is then compared to other roll numbers within the structure via an **if** within a **while loop**. An **if else** is used to print data of student with the given roll number and ask user to input new information for that student if found, else a message is shown if the inputted roll number is not found.

- **del_std: -** Primarily consists of an **if** which checks if a file exists via the **fopen** function and prints a message if the file does not exist. If it exists, the user is asked to input a roll number. An **if** within a **while loop** is implemented to compare inputted roll number to other roll numbers in the structure and if found, the respective student data is deleted via the **fclose** function which closes the respective file.

Implementation and Testing

This project is based on the management system with CRUD Operations. The Idea is to implement filling/database at the backend to store and keep the track of data. This is done by using filling and pointers. Admin has administrative privileges to manipulate any and all student data in any way they wish. Students can go through their all-time university records.

Project Breakdown Structure

The project was completed in approximately three weeks. It took longer than expected due to mid term exams and individual problems of each team member which delayed the project. All of us worked on making the functions at random while helping each other out. The third week was the most productive as we were free from mids and could work on it together. After encountering a few issues and errors, we googled them and did research which took most of the time. We worked on the project mostly on weekends as we were too exhausted after coming home from university on uni days.

Results

The result was a Student Management System with a code of about 600 lines which, when ran would produce a program that allowed a user to create student data and add it to a file, display all added data, update any inputted data and/or delete a specific student's data.

Conclusion

In conclusion, we have created a program that is above many management systems as it is precise yet simple and user friendly. Through use of different libraries, various functions that came with said libraries, various loops, and implementation of our own functions we have created a student data management system that can be used to sort any number of student data and display, update and delete it at will, the aim of the project was to learn and apply Programming Fundamentals and we have succeeded in doing that.