

FINAL REPORT

Student Attendance Calculator

Group No. - 5

GROUP MEMBERS :-

Ishaant Kumar Singh - 12203987

Shivam Shekhar - 12204084

Gaurav Pathak - 12204085

Steve S Joy - 12205035

Introduction

Student attendance calculator is an application that will help the students to calculate their attendance in each course and the aggregate attendance based on the number of based on number of lectures delivered till that week and no of lectures student have missed in a particular course. This application will help the student to have a check on his attendance for each course in order to avoid the detention from the final exams

Module Explanation

1. Login/Authentication Module:

This module will allow the user to log in and authenticate themselves to access the student attendance calculator application. The user will be required to enter their login credentials, such as a username and password, to gain access to the application. The authentication module will ensure that only authorized users can access and modify the student attendance data.

2. Insert Details of Student Module:

This module will allow the user to add new student details to the attendance calculator system. The user will be required to enter the student's name, course name, lecture attendance details, and other necessary information. The module will then store this data in the system's database for future use.

3. Display All Details Module:

This module will display all the attendance details of each student in the system. The user can view this information by selecting the display all option from the menu. The module will retrieve the data from the database and present it to the user in a readable format.

4. Search Specific Information Module:

This module will allow the user to search for a specific student's attendance information by entering their name or course name. The module will retrieve the data from the database and display it to the user in a readable format.

5. Delete/Modify Any Student Detail Module:

This module will allow the user to delete or modify the attendance details of any student in the system. The user can select the delete or modify option from the menu and enter the student's name or course name. The module will then retrieve the data from the database and allow the user to delete or modify it as required. This module will help the user to keep the attendance data up-to-date and accurate.

6. Save Data Module:

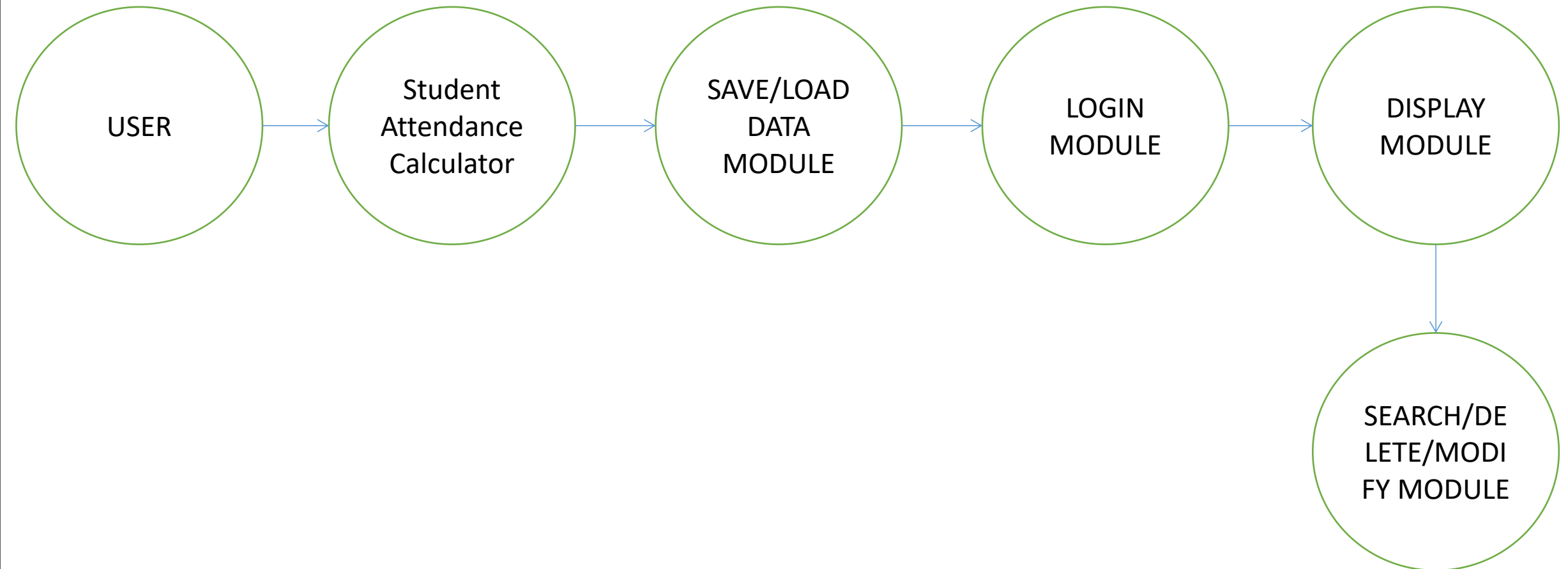
This module will allow the user to save the student attendance data to a file or database. The module will retrieve the data from the database and save it to a file in a readable format. This module will ensure that the attendance data is backed up and can be retrieved in case of any system failure or data loss.

7. Load Data Module:

This module will allow the user to load the student attendance data from a saved file or database. The user can select the load option from the menu, and the module will retrieve the data from the file and load it into the system's database. This module will ensure that the user can access and modify the attendance data even after a system restart or shutdown.

Data Flow Diagram(DFD)

Level Zero



Code Snapshots

Header Files, Main Structure and Function Initializations

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <conio.h>
5
6  #define MAX_COURSES 5
7  #define MAX_STUDENTS 100
8  #define MAX_NAME_LENGTH 10
9
10 struct student
11 {
12     int id;
13     char name[50];
14     int attendance[MAX_COURSES];
15 };
16
17 struct student students[MAX_STUDENTS];
18 int num_students = 0;
19
20 // Function prototypes
21 void login();
22 void insert_student_details();
23 void display_all_details();
24 void search_specific_information();
25 void delete_or_modify_student_detail();
26 void save_data();
27 void load_data();
```

Main Function

```
29 // Main function
30 int main()
31 {
32     load_data();
33
34     printf("Welcome To Student Attendance Calculator \n");
35     printf("Login To Proceed\n");
36
37     login();
38
39     while (1)
40     {
41         printf("\nAttendance Calculator Menu:\n");
42         printf("1. Insert details of student\n");
43         printf("2. Display all details\n");
44         printf("3. Search specific information\n");
45         printf("4. Delete/modify any student detail\n");
46         printf("5. Save and exit\n\n");
47
48         int choice;
49         printf("Enter your choice: ");
50         scanf("%d", &choice);
```



```
52     switch (choice)
53     {
54     case 1:
55         insert_student_details();
56         break;
57     case 2:
58         display_all_details();
59         break;
60     case 3:
61         search_specific_information();
62         break;
63     case 4:
64         delete_or_modify_student_detail();
65         break;
66     case 5:
67         save_data();
68         return 0;
69     default:
70         printf("Invalid choice. Please try again.\n");
71     }
72 }
73
74 return 0;
75 }
```

Login Function

```
77 void login()
78 {
79     char username[50], password[50];
80
81     printf("Enter username: ");
82     scanf("%s", username);
83     printf("Enter password: ");
84     scanf("%s", password);
85
86     if (strcmp(username, "admin") == 0 && strcmp(password, "password") == 0)
87     {
88         printf("Login successful.\n");
89     }
90     else
91     {
92         printf("Invalid username or password. Exiting...\n");
93         getch();
94         exit(0);
95     }
96 }
```

Insert Student Detail Function

```
98 void insert_student_details()
99 {
100     if (num_students >= MAX_STUDENTS)
101     {
102         printf("Maximum number of students reached. Cannot add more students.\n");
103         return;
104     }
105
106     printf("\nEnter details for student %d:\n", num_students + 1);
107     printf("Enter Registration number : ");
108     scanf("%d", &students[num_students].id);
109     printf("Enter name: ");
110     scanf("%s", students[num_students].name);
111
112     printf("Enter attendance for each course:\n");
113     for (int i = 0; i < MAX_COURSES; i++)
114     {
115         printf("Course %d: ", i + 1);
116         scanf("%d", &students[num_students].attendance[i]);
117     }
118
119     num_students++;
120
121     printf("Student details added successfully.\n");
122 }
```

Display All Details Function

```
124 void display_all_details()
125 {
126     if (num_students == 0)
127     {
128         printf("No students added yet.\n");
129         return;
130     }
131
132     // Display the header
133     printf("\n%-5s %-20s", "ID", "Name");
134     for (int i = 0; i < MAX_COURSES; i++)
135     {
136         printf(" Course %d", i + 1);
137     }
138     printf("\n");
139
140     // Display the student details
141     for (int i = 0; i < num_students; i++)
142     {
143         printf("%-5d %-20s", students[i].id, students[i].name);
144         for (int j = 0; j < MAX_COURSES; j++)
145         {
146             printf(" %8d", students[i].attendance[j]);
147         }
148         printf("\n");
149     }
150 }
```

Search Specific Information Function

```
152 void search_specific_information()
153 {
154     if (num_students == 0)
155     {
156         printf("No students added yet.\n");
157         return;
158     }
159
160     // Prompt the user to enter the student ID to search for
161     int search_id;
162     printf("Enter the student ID to search for: ");
163     scanf("%d", &search_id);
164
```



```

165 // Search for the student with the given ID
166 int found = 0;
167 for (int i = 0; i < num_students; i++)
168 {
169     if (students[i].id == search_id)
170     {
171         // Display the student details
172         printf("\nDetails for student with ID %d:\n", search_id);
173         printf("Name: %s\n", students[i].name);
174         printf("Attendance:\n");
175         for (int j = 0; j < MAX_COURSES; j++)
176         {
177             printf("Course %d: %d\n", j + 1, students[i].attendance[j]);
178         }
179         found = 1;
180         break;
181     }
182 }
183
184 if (!found)
185 {
186     printf("No student with ID %d found.\n", search_id);
187 }
188 }

```

Delete Or Modify Data Function

```
190 void delete_or_modify_student_detail()
191 {
192     // Check if there are any students
193     if (num_students == 0)
194     {
195         printf("No students added yet.\n");
196         return;
197     }
198
199     // Prompt the user to enter the student ID to delete/modify
200     int search_id;
201     printf("Enter the student ID to delete/modify: ");
202     scanf("%d", &search_id);
203
204     // Search for the student with the given ID
205     int found = 0;
206     int index;
207     for (int i = 0; i < num_students; i++)
208     {
209         if (students[i].id == search_id)
210         {
211             found = 1;
212             index = i;
213             break;
214         }
215     }
```

```

217     if (!found)
218     {
219         printf("No student with ID %d found.\n", search_id);
220         return;
221     }
222
223     // Display the student details and prompt the user for action
224     printf("\nDetails for student with ID %d:\n", search_id);
225     printf("Name: %s\n", students[index].name);
226     printf("Attendance:\n");
227     for (int j = 0; j < MAX_COURSES; j++)
228     {
229         printf("Course %d: %d\n", j + 1, students[index].attendance[j]);
230     }
231     printf("\nWhat do you want to do?\n");
232     printf("1. Delete this student\n");
233     printf("2. Modify attendance for a course\n");
234     // Prompt the user for action
235     int choice;
236     printf("Enter your choice (1 or 2): ");
237     scanf("%d", &choice);

```

```

// Student details structure
struct student {
    int id;
    char name[50];
    int attendance[MAX_COURSES];
};

// Global variables
int students[MAX_STUDENTS];
int student_count = 0;

// Function prototypes
void add_student();
void delete_student(int id);
void modify_attendance(int id, int course, int value);
void display_students();
void search_student(int id);

// Main function
int main() {
    int choice;
    do {
        printf("\nStudent Management System\n");
        printf("1. Add student\n");
        printf("2. Delete student\n");
        printf("3. Modify attendance\n");
        printf("4. Display all students\n");
        printf("5. Search student\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add_student();
                break;
            case 2:
                delete_student(-1); // Placeholder for ID
                break;
            case 3:
                modify_attendance(-1, -1, -1); // Placeholder for ID, course, value
                break;
            case 4:
                display_students();
                break;
            case 5:
                search_student(-1); // Placeholder for ID
                break;
            case 6:
                return 0;
            default:
                printf("Invalid choice\n");
        }
    } while (1);
}

// Function implementations
void add_student() {
    if (student_count == MAX_STUDENTS) {
        printf("Maximum number of students reached\n");
        return;
    }
    student s;
    s.id = student_count + 1;
    printf("Enter student name: ");
    gets(s.name);
    printf("Enter student attendance for each course (0-100):\n");
    for (int i = 0; i < MAX_COURSES; i++) {
        printf("Course %d: ", i + 1);
        scanf("%d", &s.attendance[i]);
    }
    students[student_count] = s;
    student_count++;
}

void delete_student(int id) {
    if (id == -1) {
        printf("Invalid ID\n");
        return;
    }
    for (int i = 0; i < student_count; i++) {
        if (students[i].id == id) {
            // Shift elements to the left
            for (int j = i; j < student_count - 1; j++) {
                students[j] = students[j + 1];
            }
            student_count--;
            return;
        }
    }
    printf("Student not found\n");
}

void modify_attendance(int id, int course, int value) {
    if (id == -1 || course == -1 || value == -1) {
        printf("Invalid ID, course, or value\n");
        return;
    }
    for (int i = 0; i < student_count; i++) {
        if (students[i].id == id) {
            if (course < 1 || course > MAX_COURSES) {
                printf("Invalid course\n");
                return;
            }
            if (value < 0 || value > 100) {
                printf("Invalid attendance value\n");
                return;
            }
            students[i].attendance[course - 1] = value;
            return;
        }
    }
    printf("Student not found\n");
}

void display_students() {
    if (student_count == 0) {
        printf("No students found\n");
        return;
    }
    printf("Student Details:\n");
    for (int i = 0; i < student_count; i++) {
        printf("ID: %d, Name: %s, Attendance: ", students[i].id, students[i].name);
        for (int j = 0; j < MAX_COURSES; j++) {
            printf("%d ", students[i].attendance[j]);
        }
        printf("\n");
    }
}

void search_student(int id) {
    if (id == -1) {
        printf("Invalid ID\n");
        return;
    }
    for (int i = 0; i < student_count; i++) {
        if (students[i].id == id) {
            printf("Student found: ID %d, Name %s, Attendance: ", students[i].id, students[i].name);
            for (int j = 0; j < MAX_COURSES; j++) {
                printf("%d ", students[i].attendance[j]);
            }
            printf("\n");
            return;
        }
    }
    printf("Student not found\n");
}

```



```

239     switch (choice)
240     {
241     case 1:
242         for (int i = index; i < num_students - 1; i++)
243         {
244             students[i] = students[i + 1];
245         }
246         num_students--;
247         printf("Student with ID %d deleted.\n", search_id);
248         break;
249     case 2:
250         printf("Enter the course number (1 to %d) to modify attendance for: ", MAX_COURSES);
251         int course_num;
252         scanf("%d", &course_num);
253         if (course_num < 1 || course_num > MAX_COURSES)
254         {
255             printf("Invalid course number.\n");
256             return;
257         }
258         printf("Enter the new attendance for course %d: ", course_num);
259         int new_attendance;
260         scanf("%d", &new_attendance);
261         students[index].attendance[course_num - 1] = new_attendance;
262         printf("Attendance for course %d modified for student with ID %d.\n", course_num, search_id);
263         break;
264     default:
265         printf("Invalid choice.\n");
266     }
267 }

```

Save Data Function

```
269 void save_data()
270 {
271     // Open the files for writing
272     FILE *name_file = fopen("student_names.txt", "a");
273     FILE *attendance_file = fopen("student_attendance.txt", "a");
274
275     // Write the data to the files
276     for (int i = 0; i < num_students; i++)
277     {
278         fprintf(name_file, "%d %s\n", students[i].id, students[i].name);
279         for (int j = 0; j < MAX_COURSES; j++)
280         {
281             fprintf(attendance_file, "%d ", students[i].attendance[j]);
282         }
283         fprintf(attendance_file, "\n");
284     }
285
286     fclose(name_file);
287     fclose(attendance_file);
288
289     printf("Data saved to files.\n");
290 }
```

Output Snapshots

Login Function

```
Welcome To Student Attendance Calculator  
Login To Proceed  
Enter username: admin  
Enter password: password  
Login successful.
```

After Login

```
Welcome To Student Attendance Calculator  
Login To Proceed  
Enter username: admin  
Enter password: password  
Login successful.
```

```
Attendance Calculator Menu:  
1. Insert details of student  
2. Display all details  
3. Search specific information  
4. Delete/modify any student detail  
5. Save and exit
```

```
Enter your choice: |
```

Insert Details Of Student

Attendance Calculator Menu:

1. Insert details of student
2. Display all details
3. Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 1

Enter details for student 2:

Enter Registration number : 12203987

Enter name: Ishaant

Enter attendance for each course:

Course 1: 89

Course 2: 87

Course 3: 91

Course 4: 92

Course 5: 98

Student details added successfully.

Search Specific Information

Attendance Calculator Menu:

1. Insert details of student
2. Display all details
3. Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 3

Enter the student ID to search for: 12203987

Details for student with ID 12203987:

Name: Ishaant

Attendance:

Course 1: 89

Course 2: 87

Course 3: 91

Course 4: 92

Course 5: 98

Delete Any Student Detail

Attendance Calculator Menu:

1. Insert details of student
2. Display all details
3. Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 4

Enter the student ID to delete/modify: 12203987

Details for student with ID 12203987:

Name: Ishaant

Attendance:

Course 1: 89

Course 2: 87

Course 3: 91

Course 4: 92

Course 5: 98

What do you want to do?

1. Delete this student
2. Modify attendance for a course

Enter your choice (1 or 2): 1

Student with ID 12203987 deleted.

Modify Any Student Detail

Attendance Calculator Menu:

1. Insert details of student
2. Display all details
3. Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 4

Enter the student ID to delete/modify: 12203987

Details for student with ID 12203987:

Name: Ishaant

Attendance:

Course 1: 89

Course 2: 98

Course 3: 78

Course 4: 87

Course 5: 96

What do you want to do?

1. Delete this student
2. Modify attendance for a course

Enter your choice (1 or 2): 2

Enter the course number (1 to 5) to modify attendance for: 4

Enter the new attendance for course 4: 95

Attendance for course 4 modified for student with ID 12203987.

Save Data

Attendance Calculator Menu:

1. Insert details of student
2. Display all details
3. Search specific information
4. Delete/modify any student detail
5. Save and exit

Enter your choice: 5

Data saved to files.