A

PROJECT REPORT ON

"CALENDER APPLICATION"

SUBMITTED BY:

Mr. Tikande Vedant Santosh (2124UCEF1108)

SUBJECT:

C++ PROGRAMMING

Under the guidance of

Miss. ISHWARI TIRSE

Department of Computer Science and Engineering

Sanjivani Rural Education Society's

SANJIVANI UNIVERSITY

KOPARGAON-423603, DIST: AHMEDNAGAR

2024-2025

INTRODUCTION

A University Management System is designed to simplify the administration of educational institutions. The primary goal of this system is to store, manage, and retrieve information related to students and faculty members. This project covers functionalities like adding, displaying, and searching for student and teacher records. By implementing this system in C++, we can efficiently manage large data sets while maintaining ease of use and accessibility.

CODE

```
 #include <iostream>
#include <fstream>
#include <string>

using namespace std;

// Class to manage Students
class Student {
public:
    string name;
    int rollNumber;
    string department;
    float marks;

    void addStudent() {
        cout << "Enter Student Name: ";
        cin >> name;
        cout << "Enter Roll Number: ";
        cin >> rollNumber;
        cout << "Enter Department: ";
        cin >> department;
        cout << "Enter Marks: ";
        cin >> marks;
    }

    void displayStudent() {
```

```
        cout << "Name: " << name << endl;
        cout << "Roll Number: " << rollNumber << endl;
        cout << "Department: " << department << endl;
        cout << "Marks: " << marks << endl;
    }
};

// Class to manage Teachers
class Teacher {
public:
    string name;
    int employeeId;
    string department;
    float salary;

    void addTeacher() {
        cout << "Enter Teacher Name: ";
        cin >> name;
        cout << "Enter Employee ID: ";
        cin >> employeeId;
        cout << "Enter Department: ";
        cin >> department;
        cout << "Enter Salary: ";
        cin >> salary;
    }

    void displayTeacher() {
```

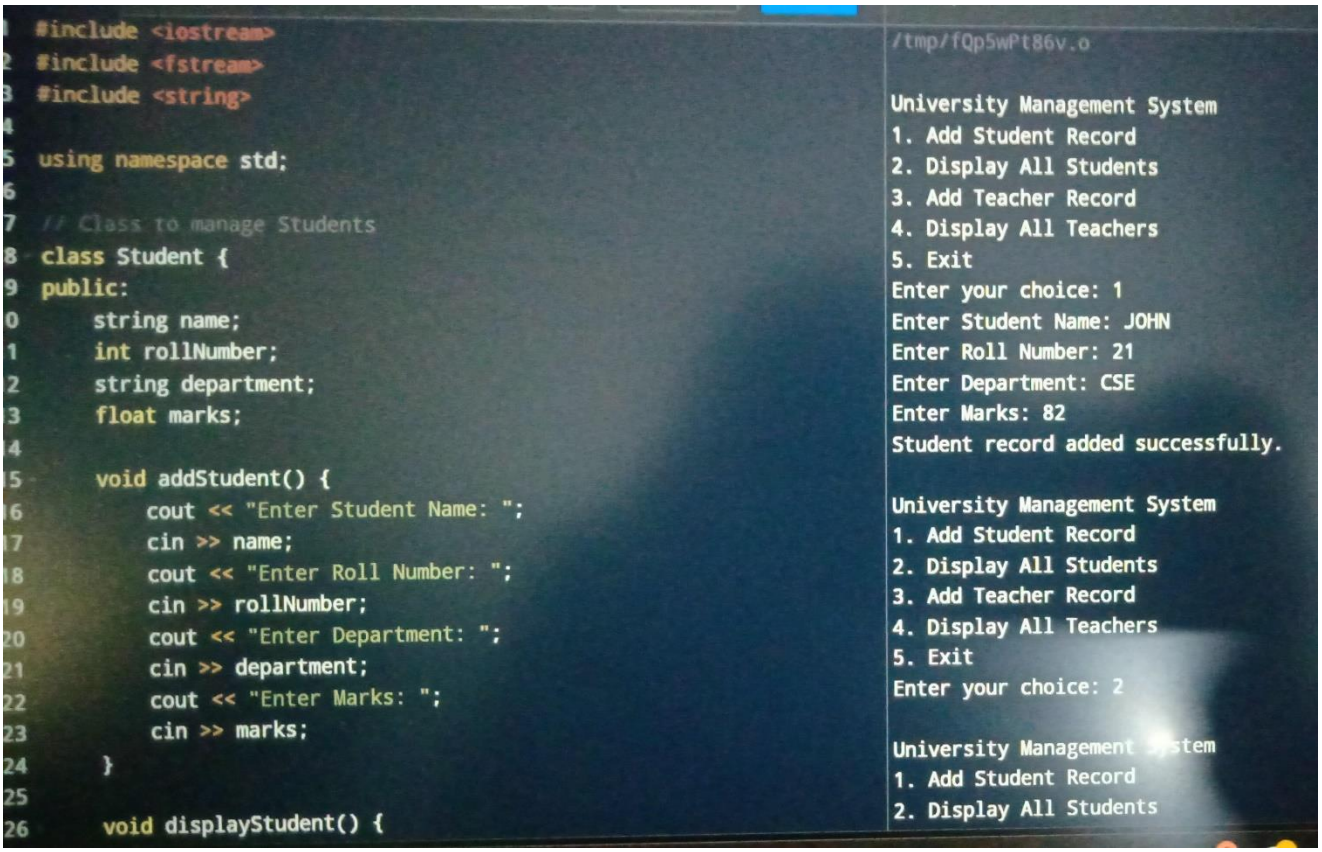
```
        cout << "Name: " << name << endl;
        cout << "Employee ID: " << employeeId << endl;
        cout << "Department: " << department << endl;
        cout << "Salary: " << salary << endl;
    }
};

// Function prototypes
void addStudentRecord();
void displayAllStudents();
void addTeacherRecord();
void displayAllTeachers();

// File handling for Students
void addStudentRecord() {
    ofstream file;
    file.open("students.txt", ios::app);
    Student student;
    student.addStudent();
    file << student.name << " " << student.rollNumber << " " << student.department << "
" << student.marks << endl;
    file.close();
    cout << "Student record added successfully.\n";
}

void displayAllStudents() {
    ifstream file;
```

```
    file.open("students.txt");
    Student student;
    while (file >> student.name >> student.rollNumber >> student.department >> student.marks) {
        student.displayStudent();
        cout << "--------------------\n";
    }
    file.close();
}

// File handling for Teacherss
void addTeacherRecord() {
    ofstream file;
    file.open("teachers.txt", ios::app);
    Teacher teacher;
    teacher.addTeacher();
    file << teacher.name << " " << teacher.employeeId << " " << teacher.department << " " << teacher.salary << endl;
    file.close();
    cout << "Teacher record added successfully.\n";
}

void displayAllTeachers() {
    ifstream file;
    file.open("teachers.txt");
    Teacher teacher;
```

```
    while (file >> teacher.name >> teacher.employeeId >> teacher.department >> teacher.salary) {
        teacher.displayTeacher();
        cout << "--------------------\n";
    }
    file.close();
}

int main() {
    int choice;
    while (true) {
        cout << "\nUniversity Management System\n";
        cout << "1. Add Student Record\n";
        cout << "2. Display All Students\n";
        cout << "3. Add Teacher Record\n";
        cout << "4. Display All Teachers\n";
        cout << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
        case 1:
            addStudentRecord();
            break;
        case 2:
            displayAllStudents();
            break;
```

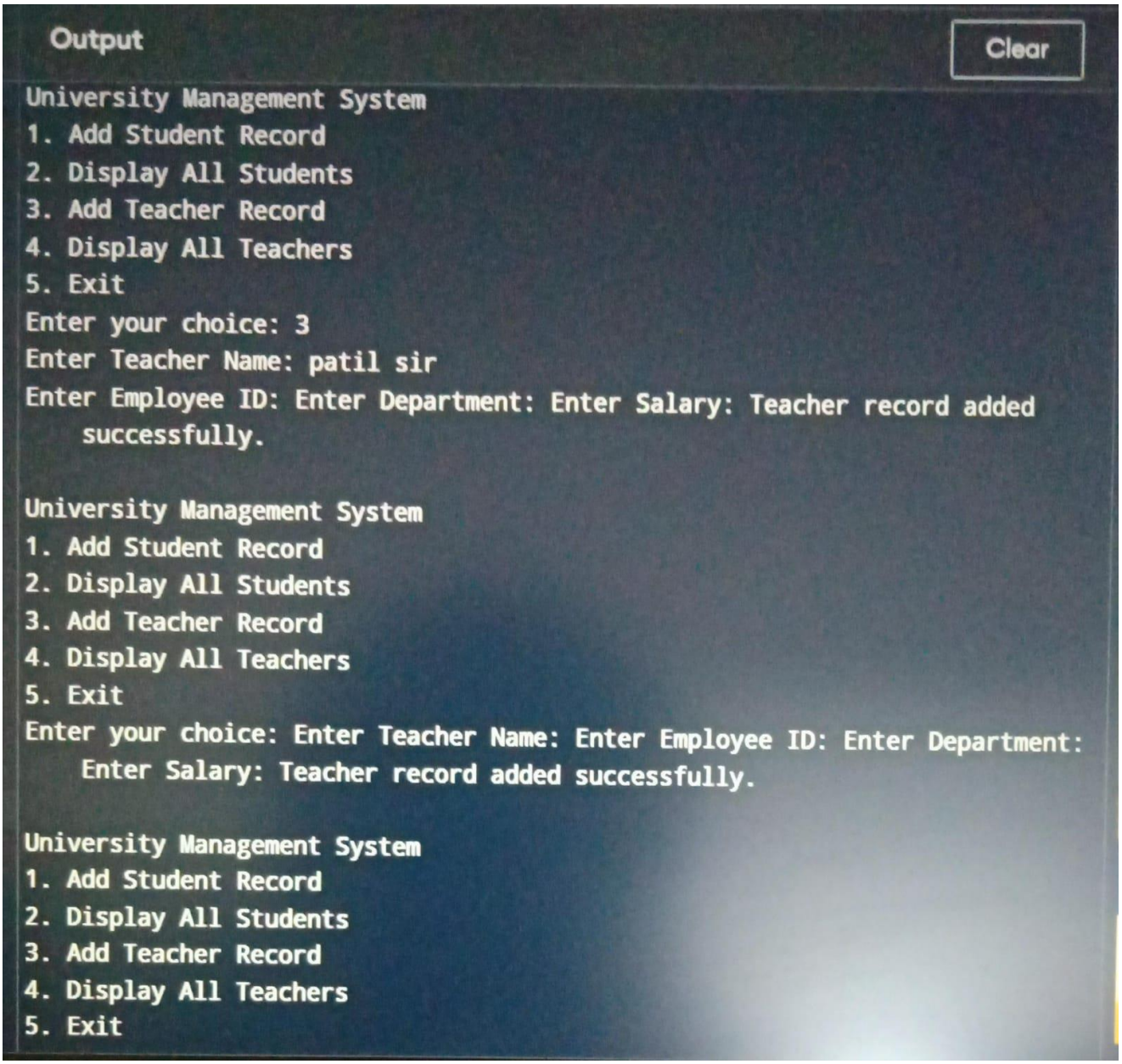
```
        case 3:
            addTeacherRecord();
            break;
        case 4:
            displayAllTeachers();
            break;
        case 5:
            cout << "Exiting...\n";
            return 0;
        default:
            cout << "Invalid choice. Please try again.\n";
        }
    }
}
```

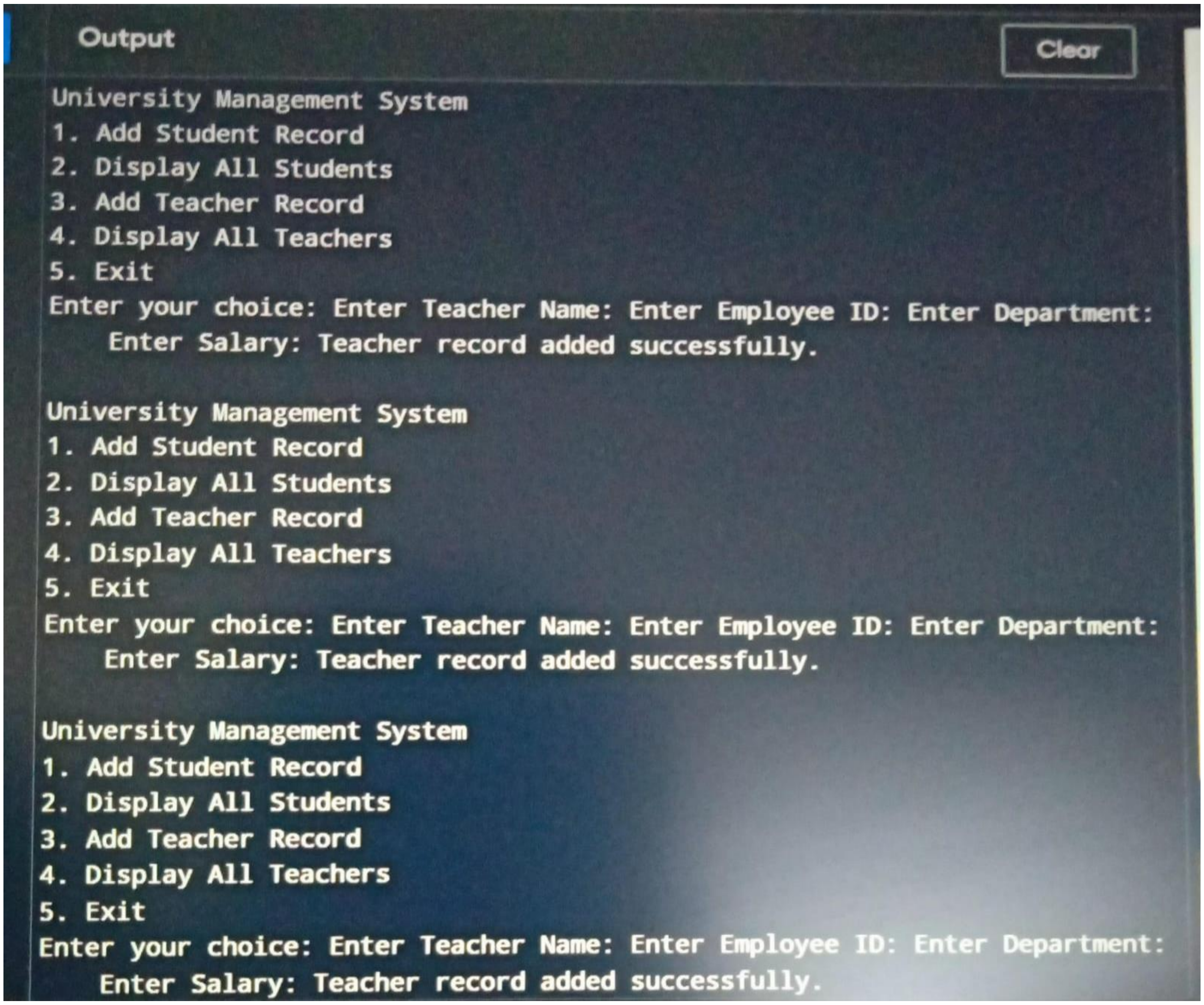
OUTPUT



```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

// Class to manage Students
class Student {
public:
    string name;
    int rollNumber;
    string department;
    float marks;

    void addStudent() {
        cout << "Enter Student Name: ";
        cin >> name;
        cout << "Enter Roll Number: ";
        cin >> rollNumber;
        cout << "Enter Department: ";
        cin >> department;
        cout << "Enter Marks: ";
        cin >> marks;
    }

    void displayStudent() {
```

```
/tmp/fQp5wPt86v.o

University Management System
1. Add Student Record
2. Display All Students
3. Add Teacher Record
4. Display All Teachers
5. Exit
Enter your choice: 1
Enter Student Name: JOHN
Enter Roll Number: 21
Enter Department: CSE
Enter Marks: 82
Student record added successfully.

University Management System
1. Add Student Record
2. Display All Students
3. Add Teacher Record
4. Display All Teachers
5. Exit
Enter your choice: 2

University Management System
1. Add Student Record
2. Display All Students
```


Output
Clear
University Management System
1. Add Student Record
2. Display All Students
3. Add Teacher Record
4. Display All Teachers
5. Exit
Enter your choice: 3
Enter Teacher Name: patil sir
Enter Employee ID: Enter Department: Enter Salary: Teacher record added successfully.
University Management System
1. Add Student Record
2. Display All Students
3. Add Teacher Record
4. Display All Teachers
5. Exit
Enter your choice: Enter Teacher Name: Enter Employee ID: Enter Department: Enter Salary: Teacher record added successfully.
University Management System
1. Add Student Record
2. Display All Students
3. Add Teacher Record
4. Display All Teachers
5. Exit


Output
Clear
University Management System
1. Add Student Record
2. Display All Students
3. Add Teacher Record
4. Display All Teachers
5. Exit
Enter your choice: Enter Teacher Name: Enter Employee ID: Enter Department: Enter Salary: Teacher record added successfully.
University Management System
1. Add Student Record
2. Display All Students
3. Add Teacher Record
4. Display All Teachers
5. Exit
Enter your choice: Enter Teacher Name: Enter Employee ID: Enter Department: Enter Salary: Teacher record added successfully.
University Management System
1. Add Student Record
2. Display All Students
3. Add Teacher Record
4. Display All Teachers
5. Exit
Enter your choice: Enter Teacher Name: Enter Employee ID: Enter Department: Enter Salary: Teacher record added successfully.

CONCLUSION

This University Management System project is a simple yet effective demonstration of managing student and teacher data using file handling in C++. It provides basic functionality like adding and displaying records, which can be extended to include more advanced features like editing and deleting records. The use of C++ concepts such as classes, file handling, and loops makes it an excellent project for learning and applying these programming fundamentals