

# Project Report

## Attendance Management System

---

## Introduction

The Attendance Management System is a console-based application developed in C++. It facilitates the management of student attendance records for educational institutions. The system provides features for both students and administrators to log in, mark attendance, view attendance records, register new students, and perform administrative tasks such as deleting students and checking attendance counts.

## Project Structure

The project consists of several source code files:

- `main.cpp`: Contains the main function and the entry point of the program. It handles user input and navigation between student and admin functionalities.
- `functions.cpp`: Contains the implementation of various functions used in the program, including user login, registration, attendance marking, and administrative tasks.
- `functions.h`: Header file containing function prototypes and necessary includes.  
    `db.dat`: A data file storing registered student usernames.

## Functionality Overview

### Main Menu

Upon launching the program, users are presented with a main menu offering options for student login, admin login, and exit.

### Student Login

Students can log in using their usernames and passwords. Upon successful login, they can mark their attendance for the current day and view their attendance count.

## **Admin Login**

Administrators can log in using predefined credentials. After logging in, they have access to various administrative functionalities:

1. Register new students by providing necessary information such as name, username, password, roll number, address, father's name, and mother's name.
2. Delete all registered students or delete a specific student by roll number.
3. View a list of all registered students along with their usernames.
4. Check the attendance count of any student by roll number.
5. Get a list of all students with their respective attendance counts.

## Working of Key Functions

### **studentLogin()**

- Prompts the user to enter their username and password.
- Calls `checkStudentCredentials()` to verify the credentials.
- Upon successful validation, displays the student menu.

### **adminLogin()**

- Asks for the admin username and password.
- Validates the credentials against predefined values.
- Grants access to admin functionalities upon successful login.

### **registerStudent()**

- Collects student information including name, username, password, roll number, address, father's name, and mother's name.
- Checks if the provided username is unique by scanning the `db.dat` file.
- If the username is unique, adds the username to `db.dat` and creates a separate data file for the student's information.
- Displays a success message upon successful registration.

### **markMyAttendance(string username)**

Allows the logged-in student to mark their attendance for the current day.

### **getListOfStudentsWithTheirPresenceCount()**

- Retrieves the list of all registered students from `db.dat`.
- For each student, reads their respective data file and calculates their attendance count.

# Conclusion

The Attendance Management System project provides a user-friendly interface for managing student attendance records efficiently. With features tailored for both students and administrators, it offers a comprehensive solution for attendance management in educational institutions. The modular design and implementation in C++ make it easily extensible and adaptable to different requirements.