

ATTENDANCE MANAGEMENT

CODE :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX 100
struct Student {
    int roll;
    char name[50];
    char dept[20];
    char attendance[10];
};

struct Student students[MAX];
int count = 0;

// Function to check alphabets only
int isAlphabet(char str[]) {
    for (int i = 0; str[i] != '\0'; i++) {
        if (!isalpha(str[i]) && str[i] != ' ') {
            return 0; // invalid
        }
    }
    return 1; // valid
}

// Check for duplicate roll numbers
int rollExists(int roll) {
    for (int i = 0; i < count; i++) {
        if (students[i].roll == roll) return 1;
    }
    return 0;
}

// Clear input buffer
void clearInputBuffer() {
    int c;
    while ((c = getchar()) != '\n' && c != EOF);
}

// Add student
void addStudent() {
    int tempRoll;
    char tempName[50], tempDept[20];

    // Roll number input with validation
    do {
        printf("Enter roll number (positive integer only): ");
        if (scanf("%d", &tempRoll) != 1) {
            printf("Invalid input. Please enter a positive integer.\n");
        }
    } while (tempRoll <= 0);
}
```

```

printf("Invalid input! Enter a positive integer.\n");
clearInputBuffer();
tempRoll = -1; // force loop to repeat
continue;
}

if (tempRoll <= 0) {
    printf("Roll number must be positive.\n");
} else if (rollExists(tempRoll)) {
    printf("Roll number already exists! Enter a unique roll number.\n");
    tempRoll = -1;
}

} while (tempRoll <= 0);
students[count].roll = tempRoll;
clearInputBuffer();

// Name input with validation
do {
    printf("Enter Name (letters and spaces only): ");
    fgets(tempName, sizeof(tempName), stdin);
    tempName[strcspn(tempName, "\n")] = 0; // remove newline

    if (!isAlphabet(tempName) || strlen(tempName) == 0) {
        printf("Invalid! Only letters and spaces allowed.\n");
    }
} while (!isAlphabet(tempName) || strlen(tempName) == 0);
strcpy(students[count].name, tempName);

// Department input with validation
do {
    printf("Enter Department (letters and spaces only): ");
    fgets(tempDept, sizeof(tempDept), stdin);
    tempDept[strcspn(tempDept, "\n")] = 0; // remove newline

    if (!isAlphabet(tempDept) || strlen(tempDept) == 0) {
        printf("Invalid! Only letters and spaces allowed.\n");
    }
} while (!isAlphabet(tempDept) || strlen(tempDept) == 0);
strcpy(students[count].dept, tempDept);

strcpy(students[count].attendance, "Not Marked");

count++;
printf("Student added successfully!\n");
}

// Mark attendance
void markAttendance() {
    int roll, found = 0;
    printf("\nEnter Roll Number to mark attendance: ");

```

```

if (scanf("%d", &roll) != 1) {
    printf("Invalid input!\n");
    clearInputBuffer();
    return;
}

for (int i = 0; i < count; i++) {
    if (students[i].roll == roll) {
        found = 1;
        char ch;
        printf("Mark Attendance for %s (P/A): ", students[i].name);
        clearInputBuffer();
        scanf(" %c", &ch);

        switch (ch) {
            case 'P':
            case 'p':
                strcpy(students[i].attendance, "Present");
                break;
            case 'A':
            case 'a':
                strcpy(students[i].attendance, "Absent");
                break;
            default:
                printf("Invalid input! Marked as Absent by default.\n");
                strcpy(students[i].attendance, "Absent");
                break;
        }
        printf("Attendance Updated!\n");
        return;
    }
}

if (!found) printf("Student not found!\n");
}

// View attendance
void viewAttendance() {
    printf("\n--- Attendance List ---\n");
    if (count == 0) {
        printf("No students added yet.\n");
        return;
    }
    for (int i = 0; i < count; i++) {
        printf("Roll: %d | Name: %s | Dept: %s | Attendance: %s\n",
               students[i].roll,
               students[i].name,
               students[i].dept,
               students[i].attendance);
    }
}

```

```
// Save to file
void saveToFile() {
    FILE *fp = fopen("attendance.txt", "w");
    if (!fp) {
        printf("Error opening file!\n");
        return;
    }
    for (int i = 0; i < count; i++) {
        fprintf(fp, "%d,%s,%s,%s\n",
                students[i].roll,
                students[i].name,
                students[i].dept,
                students[i].attendance);
    }
    fclose(fp);
    printf("\nData saved to attendance.txt successfully!\n");
}
```

```
// Main menu
int main() {
    int choice;
    while (1) {
        printf("\n--- Attendance Management System ---\n");
        printf("1. Add Student\n");
        printf("2. Mark Attendance\n");
        printf("3. View Attendance\n");
        printf("4. Save to File\n");
        printf("5. Exit\n");
        printf("Enter Your Choice: ");
        if (scanf("%d", &choice) != 1) {
            printf("Invalid input! Please enter a number 1-5.\n");
            clearInputBuffer();
            continue;
        }

        switch (choice) {
            case 1: addStudent(); break;
            case 2: markAttendance(); break;
            case 3: viewAttendance(); break;
            case 4: saveToFile(); break;
            case 5: exit(0);
            default: printf("Invalid Choice! Enter 1-5.\n");
        }
    }
    return 0;
}
```