# **Practical 1**

**Aim:** Design a console-based application in C# to track employee attendance

within an organization. The system should allow HR personnel to record

daily attendance, display all attendance entries, and calculate the number

of days an employee was present. A user-friendly console interface should

manage all interactions, with data stored in-memory during runtime

Requirements and Features:

1. Design – EmployeeAttendance

o Create a class named EmployeeAttendance to represent the

structure of a single attendance record.

o Properties:

* EmployeeId (int)
* EmployeeName (string)
* Date (DateTime)
* IsPresent (bool)

o Use constructors and encapsulation (with getters/setters)

to manage property access.

2. Data Storage and Business Logic Layer

o Maintain attendance records using a

List<EmployeeAttendance>.

2 1

o Define a separate class (e.g., AttendanceManager) that

contains methods to:

* Add a new record
* Display all records
* Calculate total present days for a specific employee

o Follow Single Responsibility Principle by separating data

operations from user interaction.

3. Console Interface and User Interaction

o Develop a menu-driven interface using do-while and switch-

case.

o Options to implement:

* Record a new attendance entry
* Display all attendance records
* View total present days by employee ID
* Exit the program

4. Input Handling and Validation

* Use try-catch blocks and conditional checks to:
* Validate date input format (DateTime.TryParse)
* Ensure numeric values for Employee ID
* Accept only true/false or Y/N for presence
* Display appropriate messages for invalid inputs.

🖥️ Sample Console Output:

===== Employee Attendance Tracker =====

1. Add Attendance Record

2. View All Attendance Records

3. Get Total Present Days by Employee ID4. Exit

Enter your choice:

**CODE:**

using System;

using System.Collections.Generic;

namespace EmployeeAttendanceTracker

{

// Model Class

public class EmployeeAttendance

{

public int EmployeeId { get; set; }

public string EmployeeName { get; set; }

public DateTime Date { get; set; }

public bool IsPresent { get; set; }

public EmployeeAttendance(int employeeId, string employeeName, DateTime date, bool isPresent)

{

EmployeeId = employeeId;

EmployeeName = employeeName;

Date = date;

IsPresent = isPresent;

}

}

// Business Logic Class

public class AttendanceManager

{

private List<EmployeeAttendance> attendanceRecords = new List<EmployeeAttendance>();

public void AddRecord(EmployeeAttendance record)

{

attendanceRecords.Add(record);

Console.WriteLine("Attendance record added successfully.\n");

}

public void DisplayAllRecords()

{

if (attendanceRecords.Count == 0)

{

Console.WriteLine("No attendance records found.\n");

return;

}

Console.WriteLine("\nAll Attendance Records:\n");

foreach (EmployeeAttendance record in attendanceRecords)

{

Console.WriteLine("ID: {0}, Name: {1}, Date: {2}, Present: {3}",

record.EmployeeId, record.EmployeeName, record.Date.ToShortDateString(), record.IsPresent);

}

Console.WriteLine();

}

public void GetTotalPresentDays(int employeeId)

{

int totalPresent = 0;

foreach (EmployeeAttendance record in attendanceRecords)

{

if (record.EmployeeId == employeeId && record.IsPresent)

{

totalPresent++;

}

}

Console.WriteLine(" Total Present Days for Employee ID {0}: {1}\n", employeeId, totalPresent);

}

}

// Console UI

class Program

{

static void Main(string[] args)

{

AttendanceManager manager = new AttendanceManager();

int choice;

do

{

Console.WriteLine("===== Employee Attendance Tracker =====");

Console.WriteLine("1. Add Attendance Record");

Console.WriteLine("2. View All Attendance Records");

Console.WriteLine("3. Get Total Present Days by Employee ID");

Console.WriteLine("4. Exit");

Console.Write("Enter your choice: ");

bool validChoice = int.TryParse(Console.ReadLine(), out choice);

if (!validChoice)

{

Console.WriteLine("Invalid input. Please enter a number between 1 and 4.\n");

continue;

}

switch (choice)

{

case 1:

AddAttendance(manager);

break;

case 2:

manager.DisplayAllRecords();

break;

case 3:

GetPresentDays(manager);

break;

case 4:

Console.WriteLine(" Exiting program. Goodbye!");

break;

default:

Console.WriteLine("Invalid choice. Try again.\n");

break;

}

} while (choice != 4);

}

static void AddAttendance(AttendanceManager manager)

{

try

{

Console.Write("Enter Employee ID (numeric): ");

string empIdInput = Console.ReadLine();

int empId;

if (!int.TryParse(empIdInput, out empId))

{

Console.WriteLine("Invalid Employee ID. Must be a number.\n");

return;

}

Console.Write("Enter Employee Name: ");

string empName = Console.ReadLine();

Console.Write("Enter Date (yyyy-MM-dd): ");

string dateInput = Console.ReadLine();

DateTime date;

if (!DateTime.TryParse(dateInput, out date))

{

Console.WriteLine("Invalid date format.\n");

return;

}

Console.Write("Was the employee present? (Y/N): ");

string presenceInput = Console.ReadLine();

bool isPresent;

if (string.Equals(presenceInput, "Y", StringComparison.OrdinalIgnoreCase))

{

isPresent = true;

}

else if (string.Equals(presenceInput, "N", StringComparison.OrdinalIgnoreCase))

{

isPresent = false;

}

else

{

Console.WriteLine("Invalid input. Enter Y or N.\n");

return;

}

EmployeeAttendance record = new EmployeeAttendance(empId, empName, date, isPresent);

manager.AddRecord(record);

}

catch (Exception ex)

{

Console.WriteLine("Error: " + ex.Message + "\n");

}

}

static void GetPresentDays(AttendanceManager manager)

{

Console.Write("Enter Employee ID to check: ");

string empIdInput = Console.ReadLine();

int empId;

if (int.TryParse(empIdInput, out empId))

{

manager.GetTotalPresentDays(empId);

}

else

{

Console.WriteLine("Invalid Employee ID. Must be a number.\n");

}

}

}

}

**OUTPUT:**

