

**Attendance Management System**

HAMIM MD NUR(云舒)

2111551204

Net Programming

Chen Laoshi

8 on Nov, 2024

**Table OF Contents**

**[1.2 Abstract:](#_Toc181744602)** [3](#_Toc181744602)

**[1.3 Introduction:](#_Toc181744603)** [3](#_Toc181744603)

**[1.4 Main Body:](#_Toc181744604)** [4](#_Toc181744604)

**[Chapter 2: System Requirements and Analysis](#_Toc181744605)** [4](#_Toc181744605)

**[Chapter 3: System Design and Architecture](#_Toc181744606)** [5](#_Toc181744606)

**[Chapter 4: Implementation and Testing](#_Toc181744607)** [6](#_Toc181744607)

**[1.4.2 Screenshots Of the Program & Short Description:](#_Toc181744608)** [7](#_Toc181744608)

**[1.5 Conclusion](#_Toc181744609)** [46](#_Toc181744609)

**[1.5.1 Main Findings](#_Toc181744610)** [46](#_Toc181744610)

**[1.5.2 Value and Alternative Insights](#_Toc181744611)** [46](#_Toc181744611)

**[1.5.3 Future Possibilities](#_Toc181744612)** [46](#_Toc181744612)

# **1.2 Abstract:**

This report details the creation and implementation of an Attendance Management System (AMS) designed to improve the efficiency of tracking employee attendance within an organization. The system offers features for logging, retrieving, and updating attendance records, and it accommodates a range of statuses such as late arrival, early leave, absence, and full attendance. Developed as a Windows Forms application, it provides a user-friendly interface that enables searches by employee ID, name, or attendance status. By digitizing attendance management, the system enhances data reliability and accessibility, minimizing the risks of errors commonly associated with manual methods. This project follows a systematic process, from requirements analysis to design, coding, and testing, resulting in a comprehensive solution tailored to attendance tracking needs.

# **1.3 Introduction:**

In a professional setting, keeping precise attendance records is vital for effective employee management and maintaining productivity. This project, the Attendance Management System (AMS), is designed to overcome the common challenges of tracking attendance. It offers an intuitive interface that simplifies the processes of recording and viewing attendance data, ensuring ease of use and reliability.

1. **Research Question or Purpose:** The purpose of this project is to develop an automated Attendance Management System that simplifies record-keeping, enables efficient data retrieval, and enhances data accuracy for tracking employee attendance.
2. **Justification (Significance):** Accurate attendance tracking is essential for managing employee productivity, determining payroll, and maintaining organizational accountability. Manual tracking systems are prone to errors, time-consuming, and challenging to update. By providing a streamlined, automated solution, this Attendance Management System reduces the administrative burden on HR staff, minimizes errors, and allows quick and efficient access to attendance records.
3. **Background and Literature Review:** In most organizations, attendance management is critical for tracking employee productivity and calculating payroll. Traditional methods, such as paper-based or spreadsheet-based systems, lack the efficiency and accuracy required in modern workplaces. Existing attendance management systems provide some automation but often fail to cover specific needs, such as tracking multiple attendance statuses (e.g., tardiness, absenteeism) and providing secure options for modifying records. This project addresses these limitations by developing a Windows Forms application that supports various attendance statuses, secure data modification, and a user-friendly search interface.
4. **Sources of Data and Methodology:**

* **Data Sources:** The system was tested using sample employee records created for simulation. These records included employee ID, name, check-in and check-out times, and status flags for tardiness, early departure, absenteeism, and full attendance.
* **Methodology:**

**Form-Based Interface Design:** The system was developed using Windows Forms, allowing a user-friendly interface that simplifies navigation and data entry.

**Data Handling:** Each attendance record includes fields for Employee ID, name, and attendance statuses. The system is designed to retrieve records based on flexible search criteria, supporting HR needs for quick access to specific attendance information.

1. **Organization of the Paper:**

The paper is organized as follows:

* **Chapter 2:** System Requirements and Analysis, outlining the functional and non-functional requirements.
* **Chapter 3:** System Design and Architecture, detailing the design choices and form layout.
* **Chapter 4:** Implementation and Testing, explaining the core functionalities and the testing process.
* **Chapter 5:** Conclusion, summarizing findings, highlighting system value, and suggesting future extensions.

# **1.4 Main Body:**

## **Chapter 2: System Requirements and Analysis**

## 

**2.1** **Functional Requirements**

* The system must allow attendance records to be added with fields for Employee ID, name, and attendance statuses (tardiness, early departure, absenteeism, and full attendance).
* Users must be able to search attendance records by Employee ID, name, tardiness, early departure, absenteeism, and full attendance status.
* The system must support modifying or deleting attendance records through a password-protected interface.

**2.2 Non-Functional Requirements**

* The system must have an intuitive user interface, developed as a Windows Forms application, allowing ease of use and accessibility.
* Data accuracy and consistency must be maintained throughout all operations.
* System performance must allow rapid searching, retrieval, and display of attendance data, even for large datasets.

## **Chapter 3: System Design and Architecture**

**3.1 Design Choices**

This form provides secure access to the system. The application supports one type of users (e.g., Admin) and validates credentials through encrypted storage. Passwords are hidden during entry, with options for saving user credentials for returning users.

**3.2 Form Layouts and Data Handling**

After a successful login, the main form provides access to various attendance management features. It includes buttons for adding, modifying, viewing, and searching attendance records, allowing for easy navigation and task completion. Additionally, an Exit Button enables users to close the application securely from the main dashboard.

**The main forms include:**

1. **Attendance Input:** Users can add attendance records by entering the check-in and check-out times. The system calculates tardiness and early departures based on predefined working hours.
2. **View Attendance:** This feature displays a summary of attendance records in a DataGridView. It includes fields for employee ID, name, check-in/check-out times, tardiness, early departures, and absenteeism status.
3. **Modify Attendance:** Allows admins to modify existing attendance records, with password validation for secure access. The form allows changes to check-in, check-out times, tardiness, and absenteeism fields.
4. **Delete Attendance:** Securely deletes attendance records using password validation.
5. **Exit Button:** In addition to other functionalities, the main form includes an Exit Button for quickly closing the application, ensuring ease of use and a safe exit from the system.

**3.3 Data Flow and Processing**

The system follows a structured data flow:

* **Input**: Users enter employee attendance details or search criteria in the respective forms.
* **Processing**: The system processes search or modification requests by filtering the dataset and applying attendance status criteria.
* **Output**: The results display in a DataGridView, and users can select specific records to view or modify as needed.

## **Chapter 4: Implementation and Testing**

**4.1 Implementation of Core Functionalities**

1. **Adding Attendance Records**: Users input attendance data from dropdowns, click the two buttons for check-in and check-out times. The system calculates tardiness and early departures based on predefined working hours.
2. **Searching Attendance**: The Search Attendance form allows searching by Employee ID and name with filters for attendance statuses. Results display in a DataGridView, and users can select a record for more detailed viewing or editing.
3. **Modifying and Deleting Records**: A password-protected Modify/Delete form permits secure editing of attendance records. Only authorized users can alter existing records, ensuring data security.

**4.2 Testing Procedures and Results**

The system was rigorously tested with sample attendance records to ensure all

functionalities worked as expected:

* **Functional Testing:**  Verified each form (Add, Search, View, Modify/Delete) to ensure correct data handling and display.
* **Usability Testing:** Ensured the interface was user-friendly, and all functionalities were easily accessible.
* **Security Testing:** Tested password protection for modifying and deleting records to prevent unauthorized access.

Issues discovered during testing were addressed promptly, including resolving data

consistency problems with attendance status display in search results.

## **1.4.2 Screenshots Of the Program & Short Description:**

**Start Page Code:**

**namespace AttendanceManagementSystem**

**{**

**public partial class LoadingForm : Form**

**{**

**public LoadingForm()**

**{**

**InitializeComponent();**

**}**

**private void timer1\_Tick(object sender, EventArgs e)**

**{**

**panel2.Width += 3;**

**if (panel2.Width >= 599)**

**{**

**timer1.Stop();**

**this.Close(); // Close the loading form without opening the login form here**

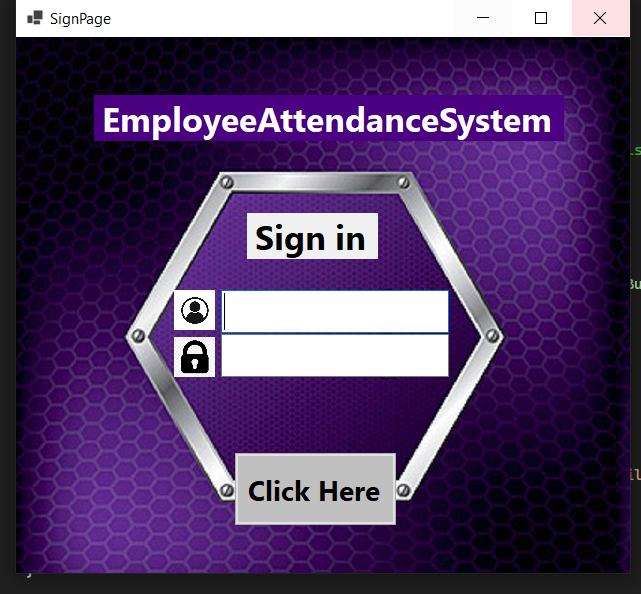
**}**

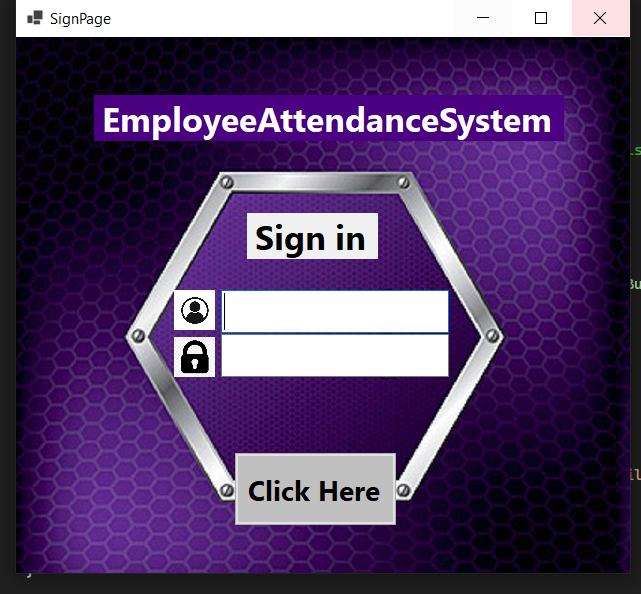
**}**

**}**

**}**

**Login page:** Validates user credentials and provides secure system access.





**Sign In Page Code:**

**namespace EmployeeAttendancceManagementSystem**

**{**

**public partial class SignInPage : Form**

**{**

**public SignInPage()**

**{**

**InitializeComponent();**

**}**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**// Replace these with actual validation checks or database calls**

**string username = "admin"; // Example username**

**string password = "0000"; // Example password**

**if (textBox1.Text == username && textBox3.Text == password)**

**{**

**// Successful login**

**MessageBox.Show("Login successful!", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**// Open the MainForm and close the login form**

**Mainform mainForm = new Mainform();**

**mainForm.Show();**

**this.Hide();**

**}**

**else**

**{**

**// Invalid login**

**MessageBox.Show("Invalid username or password.", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**textBox1.Clear();**

**textBox3.Focus();**

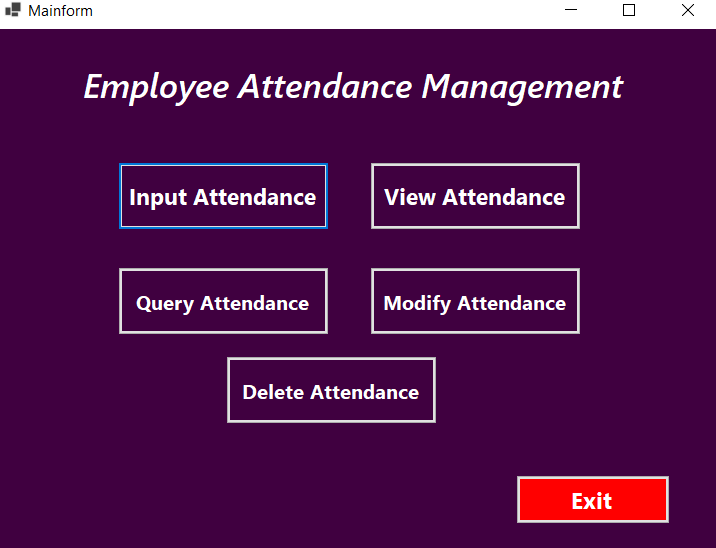
**}**

**}**

**}**

**}**

**Main Dashboard:** The main navigation hub for all attendance management functions.



**Main Dashboard Form Code:**

**namespace AttendanceManagementSystem**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**namespace EmployeeAttendancceManagementSystem**

**{**

**public partial class Mainform : Form**

**{**

**public Mainform()**

**{**

**InitializeComponent();**

**this.KeyPreview = true;**

**this.KeyDown += MainForm\_KeyDown;**

**// Manually assign button click events if not wired in designer**

**button1.Click += button1\_Click;**

**button2.Click += button2\_Click;**

**button3.Click += button3\_Click;**

**button4.Click += button4\_Click;**

**button5.Click += button5\_Click;**

**button6.Click += button6\_Click;**

**}**

**// Handle KeyDown event for selecting functions with keys 1 to 5 and exiting with 6**

**private void MainForm\_KeyDown(object? sender, KeyEventArgs e)**

**{**

**switch (e.KeyCode)**

**{**

**case Keys.D1: // Press 1 to open Input Attendance Form**

**OpenInputAttendanceForm();**

**break;**

**case Keys.D2: // Press 2 to open View Attendance Form**

**OpenViewAttendanceForm();**

**break;**

**case Keys.D3: // Press 3 to open Search Attendance Form**

**OpenQueryAttendanceForm();**

**break;**

**case Keys.D4: // Press 4 to open Modify Attendance Form**

**OpenModifyAttendanceForm();**

**break;**

**case Keys.D5: // Press 5 to open Delete Attendance Form**

**OpenDeleteAttendanceForm();**

**break;**

**case Keys.D6: // Press 6 to exit the application**

**Application.Exit();**

**break;**

**default:**

**MessageBox.Show("Invalid input! Please enter a number between 1 and 6.");**

**break;**

**}**

**}**

**// Methods to open each form**

**private void OpenInputAttendanceForm()**

**{**

**InputAttendanceForm inputForm = new InputAttendanceForm();**

**inputForm.ShowDialog(); // Open input Attendance Form**

**}**

**private void OpenViewAttendanceForm()**

**{**

**ViewAttendanceForm viewForm = new ViewAttendanceForm();**

**viewForm.ShowDialog(); // Open View Attendance Form**

**}**

**private void OpenQueryAttendanceForm()**

**{**

**QueryAttendanceForm queryForm = new QueryAttendanceForm();**

**queryForm.ShowDialog(); // Open query Attendance Form**

**}**

**private void OpenModifyAttendanceForm()**

**{**

**ModifyAttendanceForm modifyForm = new ModifyAttendanceForm();**

**modifyForm.ShowDialog(); // Open Modify Attendance Form**

**}**

**private void OpenDeleteAttendanceForm()**

**{**

**DeleteAttendanceForm deleteForm = new DeleteAttendanceForm();**

**deleteForm.ShowDialog(); // Open Delete Attendance Form**

**}**

**// Button click event handlers**

**private void button1\_Click(object? sender, EventArgs e)**

**{**

**OpenInputAttendanceForm();**

**}**

**private void button2\_Click(object? sender, EventArgs e)**

**{**

**OpenViewAttendanceForm();**

**}**

**private void button3\_Click(object? sender, EventArgs e)**

**{**

**OpenQueryAttendanceForm();**

**}**

**private void button4\_Click(object? sender, EventArgs e)**

**{**

**OpenModifyAttendanceForm();**

**}**

**private void button5\_Click(object? sender, EventArgs e)**

**{**

**OpenDeleteAttendanceForm();**

**}**

**private void button6\_Click(object? sender, EventArgs e)**

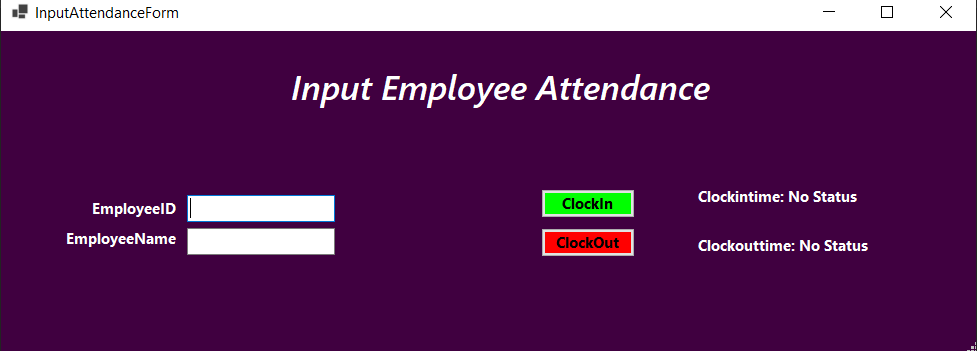
**{**

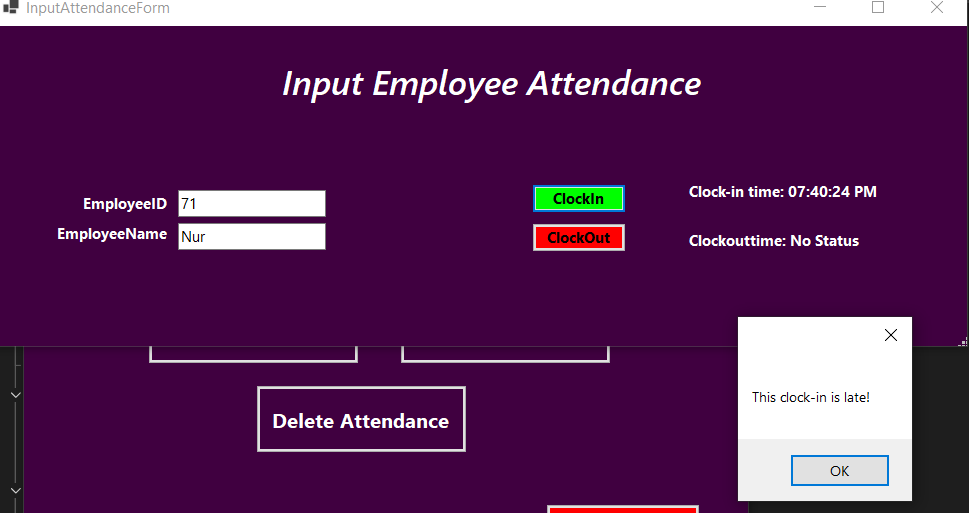
**Application.Exit();**

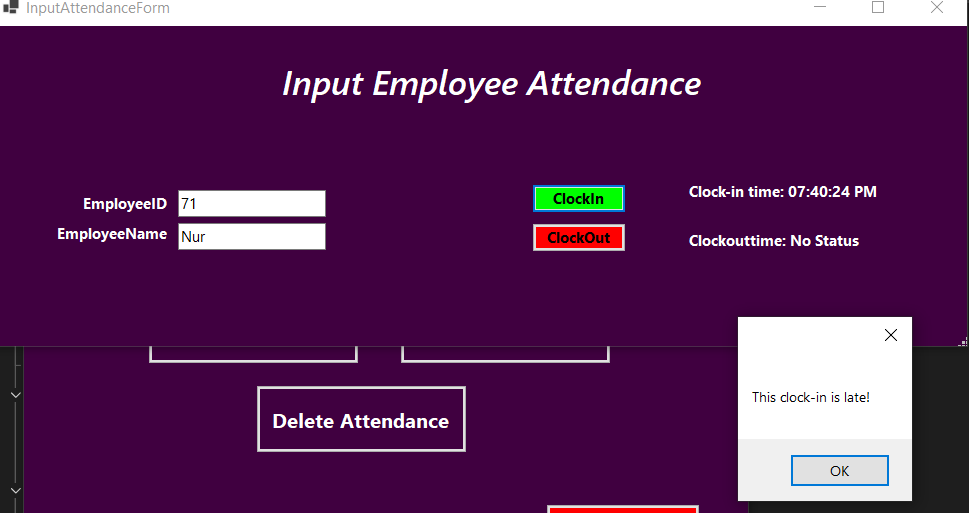
**}**

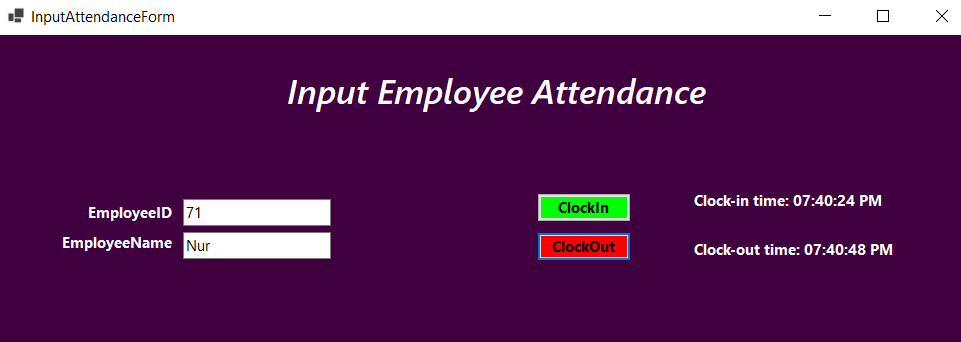
**}**

**}**









**Input Attendance Form Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**namespace EmployeeAttendancceManagementSystem**

**{**

**public partial class InputAttendanceForm : Form**

**{**

**public InputAttendanceForm()**

**{**

**InitializeComponent();**

**}**

**private (int, string) GetSelectedEmployeeDetails()**

**{**

**if (int.TryParse(textBox1.Text, out int employeeID) && !string.IsNullOrWhiteSpace(textBox2.Text))**

**{**

**return (employeeID, textBox2.Text);**

**}**

**MessageBox.Show("Please enter a valid Employee ID and Employee Name.");**

**return (-1, string.Empty);**

**}**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**var (employeeID, employeeName) = GetSelectedEmployeeDetails();**

**if (employeeID == -1 || employeeName == null)**

**return;**

**var employee = EmployeeData.Employees.FirstOrDefault(emp => emp.EmployeeID == employeeID && emp.EmployeeName.Equals(employeeName, StringComparison.OrdinalIgnoreCase));**

**if (employee != null)**

**{**

**employee.ClockInTime = DateTime.Now;**

**label4.Text = "Clock-in time: " + employee.ClockInTime?.ToString("hh:mm:ss tt");**

**// Define start time as 8:00 AM**

**var startTime = new TimeSpan(8, 0, 0);**

**// Check if clock-in time is after 8:00 AM**

**if (employee.ClockInTime.HasValue && employee.ClockInTime.Value.TimeOfDay > startTime)**

**{**

**employee.Tardiness = 1;**

**MessageBox.Show("This clock-in is late!");**

**}**

**else**

**{**

**employee.Tardiness = 0;**

**}**

**}**

**else**

**{**

**MessageBox.Show("Employee not found!");**

**}**

**}**

**private void button2\_Click(object sender, EventArgs e)**

**{**

**var (employeeID, employeeName) = GetSelectedEmployeeDetails();**

**if (employeeID == -1 || employeeName == null)**

**return;**

**var employee = EmployeeData.Employees.FirstOrDefault(emp => emp.EmployeeID == employeeID && emp.EmployeeName.Equals(employeeName, StringComparison.OrdinalIgnoreCase));**

**if (employee != null)**

**{**

**employee.ClockOutTime = DateTime.Now;**

**label5.Text = "Clock-out time: " + employee.ClockOutTime?.ToString("hh:mm:ss tt");**

**// Define end time as 5:30 PM**

**var endTime = new TimeSpan(17, 30, 0);**

**// Check if clock-out time is before 5:30 PM**

**if (employee.ClockOutTime.HasValue && employee.ClockOutTime.Value.TimeOfDay < endTime)**

**{**

**employee.EarlyDepartures = 1;**

**MessageBox.Show("This clock-out is early!");**

**}**

**else**

**{**

**employee.EarlyDepartures = 0;**

**}**

**}**

**else**

**{**

**MessageBox.Show("Employee not found!");**

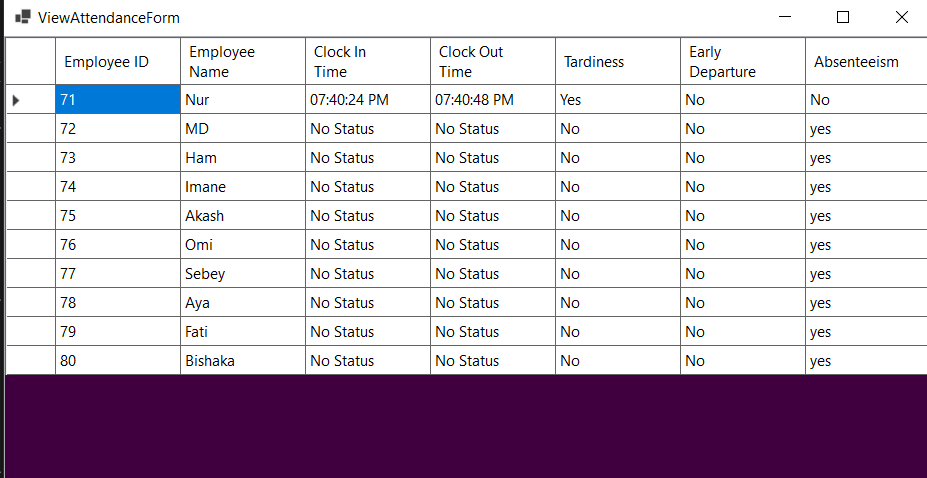
**}**

**}**

**}**

**}**

**Input Attendance**: Interface for recording new attendance entries.



**View Attendance Form Code:**

**View Attendance Form Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**namespace EmployeeAttendancceManagementSystem**

**{**

**public partial class ViewAttendanceForm : Form**

**{**

**public ViewAttendanceForm()**

**{**

**InitializeComponent();**

**InitializeDataGridView();**

**// Check absenteeism status before loading records**

**EmployeeData.EndOfDayCheck();**

**LoadAttendanceRecords();**

**}**

**private void InitializeDataGridView()**

**{**

**dataGridView1.AutoGenerateColumns = false;**

**dataGridView1.Columns.Clear();**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Employee ID",**

**DataPropertyName = "EmployeeID",**

**Name = "EmployeeIDColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Employee Name",**

**DataPropertyName = "EmployeeName",**

**Name = "EmployeeNameColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Clock In Time",**

**DataPropertyName = "CheckIn", // Match with anonymous object in LoadAttendanceRecords**

**Name = "ClockinTimeColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Clock Out Time",**

**DataPropertyName = "CheckOut", // Match with anonymous object in LoadAttendanceRecords**

**Name = "ClockoutTimeColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Tardiness",**

**DataPropertyName = "Tardiness", // No change needed**

**Name = "TardinessColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Early Departure",**

**DataPropertyName = "EarlyDepartures", // Match with anonymous object in LoadAttendanceRecords**

**Name = "EarlyDeparturesColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Absenteeism",**

**DataPropertyName = "Absenteeism",**

**Name = "AbsenteeismColumn"**

**});**

**dataGridView1.Dock = DockStyle.Fill;**

**dataGridView1.BorderStyle = BorderStyle.Fixed3D;**

**dataGridView1.ColumnHeadersHeightSizeMode = DataGridViewColumnHeadersHeightSizeMode.AutoSize;**

**dataGridView1.Margin = new Padding(10);**

**}**

**private void ViewAttendanceForm\_Load(object sender, EventArgs e)**

**{**

**// Create a layout panel for organizing the form**

**var panel = new TableLayoutPanel**

**{**

**Dock = DockStyle.Fill,**

**ColumnCount = 1,**

**RowCount = 2,**

**Padding = new Padding(10),**

**AutoSize = true**

**};**

**}**

**private void LoadAttendanceRecords()**

**{**

**var attendanceRecords = EmployeeData.Employees**

**.Select(emp => new**

**{**

**EmployeeID = emp.EmployeeID,**

**EmployeeName = emp.EmployeeName,**

**CheckIn = emp.ClockInTime.HasValue ? emp.ClockInTime.Value.ToString("hh:mm:ss tt") : "No Status",**

**CheckOut = emp.ClockOutTime.HasValue ? emp.ClockOutTime.Value.ToString("hh:mm:ss tt") : "No Status",**

**Tardiness = emp.Tardiness == 1 ? "Yes" : "No",**

**EarlyDepartures = emp.EarlyDepartures == 1 ? "Yes" : "No",**

**Absenteeism = emp.Absenteeism == 1 ? "yes" : "No"**

**})**

**.ToList();**

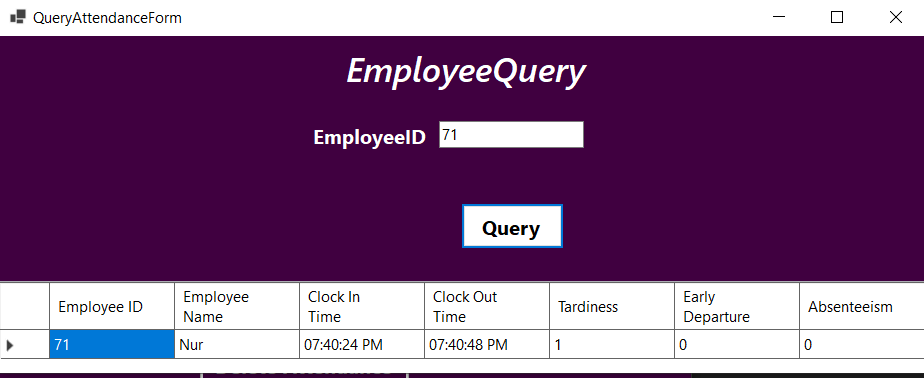
**dataGridView1.DataSource = attendanceRecords;**

**}**

**}**

**}**

**Query Attendance:**



**Employee Query Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**namespace EmployeeAttendancceManagementSystem**

**{**

**public partial class QueryAttendanceForm : Form**

**{**

**public QueryAttendanceForm()**

**{**

**InitializeComponent();**

**InitializeDataGridView();**

**}**

**private void InitializeDataGridView()**

**{**

**dataGridView1.AutoGenerateColumns = false;**

**dataGridView1.Columns.Clear();**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Employee ID",**

**DataPropertyName = "EmployeeID",**

**Name = "EmployeeIDColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Employee Name",**

**DataPropertyName = "EmployeeName",**

**Name = "EmployeeNameColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Clock In Time",**

**DataPropertyName = "CheckIn", // Match with anonymous object in LoadAttendanceRecords**

**Name = "ClockinTimeColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Clock Out Time",**

**DataPropertyName = "CheckOut", // Match with anonymous object in LoadAttendanceRecords**

**Name = "ClockoutTimeColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Tardiness",**

**DataPropertyName = "Tardiness", // No change needed**

**Name = "TardinessColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Early Departure",**

**DataPropertyName = "EarlyDepartures", // Match with anonymous object in LoadAttendanceRecords**

**Name = "EarlyDeparturesColumn"**

**});**

**dataGridView1.Columns.Add(new DataGridViewTextBoxColumn**

**{**

**HeaderText = "Absenteeism",**

**DataPropertyName = "Absenteeism",**

**Name = "AbsenteeismColumn"**

**});**

**// Optional: Customize appearance**

**dataGridView1.BorderStyle = BorderStyle.Fixed3D;**

**dataGridView1.ColumnHeadersHeightSizeMode = DataGridViewColumnHeadersHeightSizeMode.AutoSize;**

**}**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**// Retrieve input from TextBoxes**

**string employeeId = textBox1.Text.Trim();**

**// Perform search based on Employee ID**

**var filteredRecords = EmployeeData.Employees**

**.Where(emp =>**

**(string.IsNullOrEmpty(employeeId) || emp.EmployeeID.ToString() == employeeId)**

**)**

**.Select(emp => new**

**{**

**emp.EmployeeID,**

**EmployeeName = emp.EmployeeName,**

**CheckIn = emp.ClockInTime?.ToString("hh:mm:ss tt") ?? "No Status",**

**CheckOut = emp.ClockOutTime?.ToString("hh:mm:ss tt") ?? "No Status",**

**emp.Tardiness,**

**emp.EarlyDepartures,**

**emp.Absenteeism**

**})**

**.ToList();**

**// Display results in DataGridView**

**if (filteredRecords.Any())**

**{**

**dataGridView1.DataSource = filteredRecords;**

**}**

**else**

**{**

**MessageBox.Show("No records match the query criteria.", "Query Results", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**dataGridView1.DataSource = null; // Clear grid if no results**

**}**

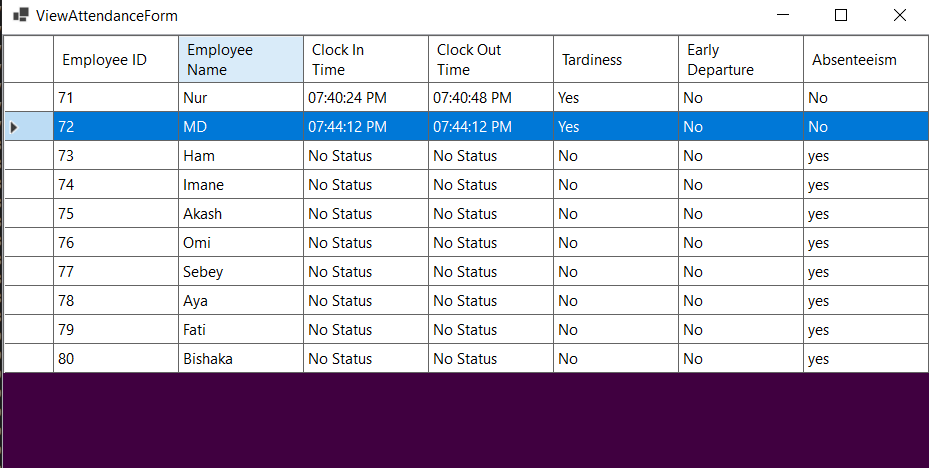
**}**

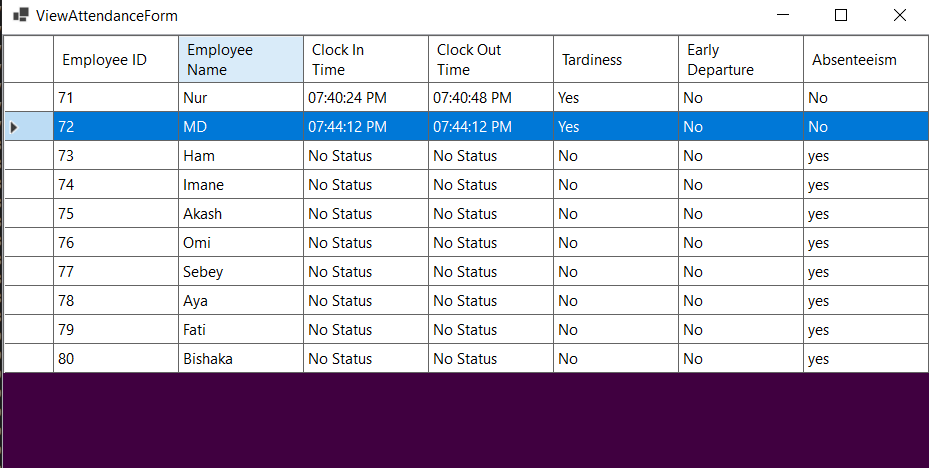
**}**

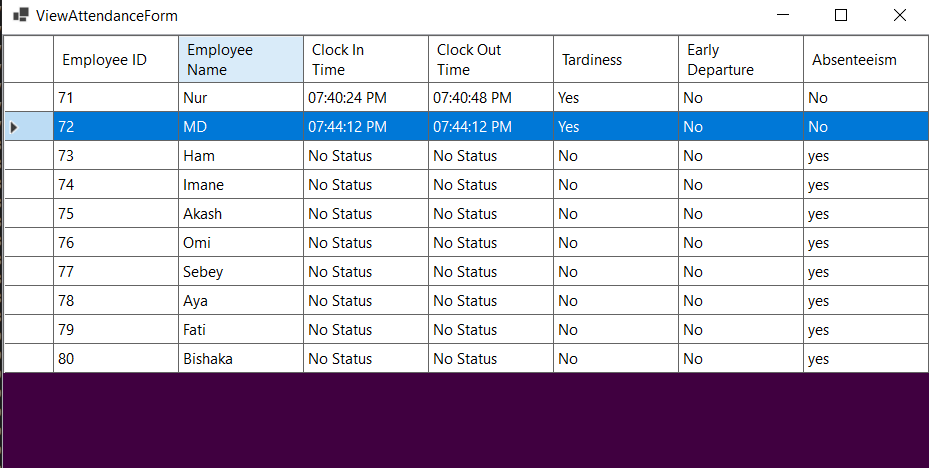
**}**

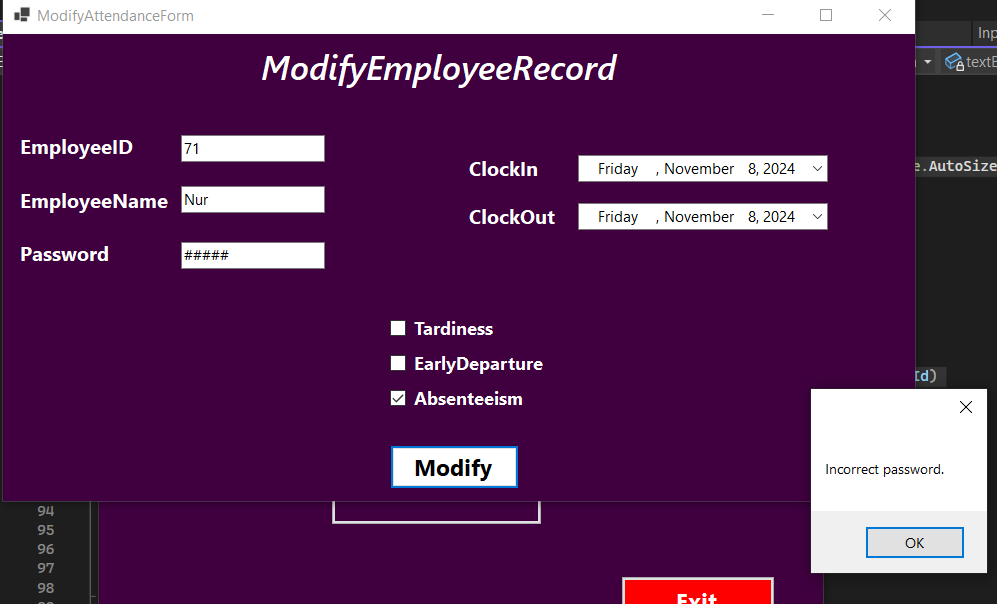
**Modify Attendance:**











**Modify Attendance Form Code:**

**Modify Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**namespace EmployeeAttendancceManagementSystem**

**{**

**public partial class ModifyAttendanceForm : Form**

**{**

**private const string AdminPassword = "0000"; // Set admin password here**

**public ModifyAttendanceForm()**

**{**

**InitializeComponent();**

**}**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**// Check if password is empty**

**if (string.IsNullOrEmpty(textBox3.Text))**

**{**

**MessageBox.Show("Please enter the password.");**

**return;**

**}**

**if (textBox3.Text != AdminPassword)**

**{**

**MessageBox.Show("Incorrect password.");**

**return;**

**}**

**var employee = EmployeeData.Employees**

**.FirstOrDefault(emp =>**

**emp.EmployeeID.ToString() == textBox1.Text &&**

**emp.EmployeeName == textBox2.Text);**

**if (employee == null)**

**{**

**MessageBox.Show("Employee record not found.");**

**return;**

**}**

**// Update fields based on user input**

**employee.ClockInTime = dateTimePicker1.Value;**

**employee.ClockOutTime = dateTimePicker2.Value;**

**// Update Tardiness and Early Departures based on checkbox states**

**employee.Tardiness = checkBox1.Checked ? 1 : 0;**

**employee.EarlyDepartures = checkBox2.Checked ? 1 : 0;**

**// Handle Absenteeism**

**if (checkBox3.Checked)**

**{**

**// Mark as Absent if Absenteeism checkbox is checked**

**employee.SetAbsenteeism(1);**

**}**

**else**

**{**

**// If unchecked, mark as Present (ensure both Check-In and Check-Out are set)**

**if (employee.ClockInTime.HasValue && employee.ClockOutTime.HasValue)**

**{**

**employee.SetAbsenteeism(0); // Present**

**}**

**else**

**{**

**MessageBox.Show("Please provide both Clock-In and Clock-Out times to mark as Present.");**

**return;**

**}**

**}**

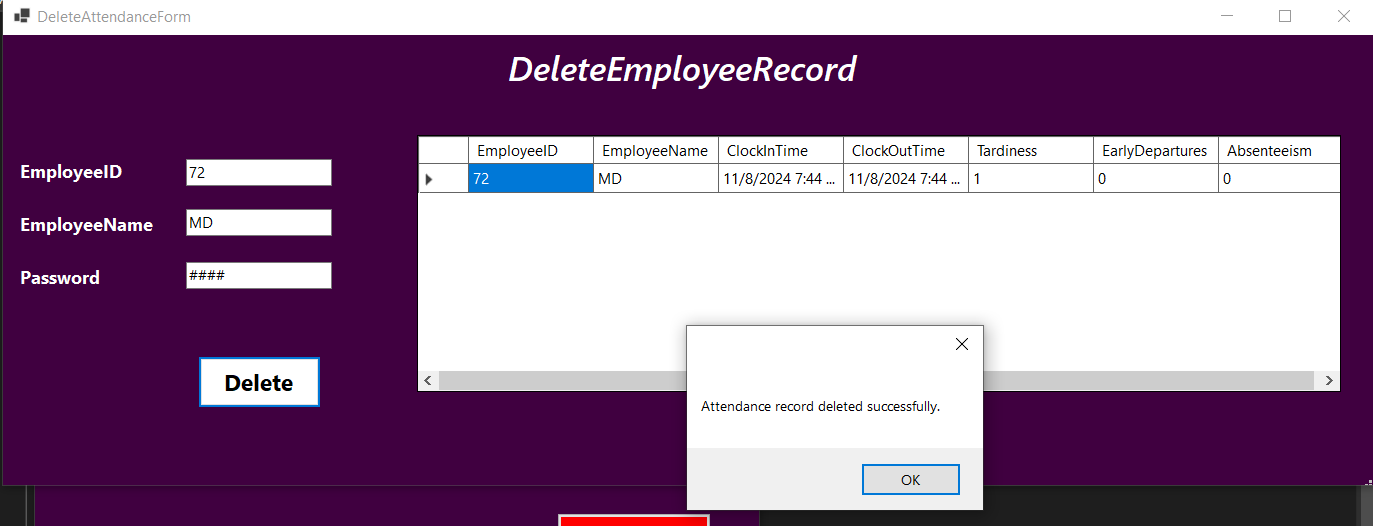
**MessageBox.Show("Attendance record modified successfully.");**

**}**

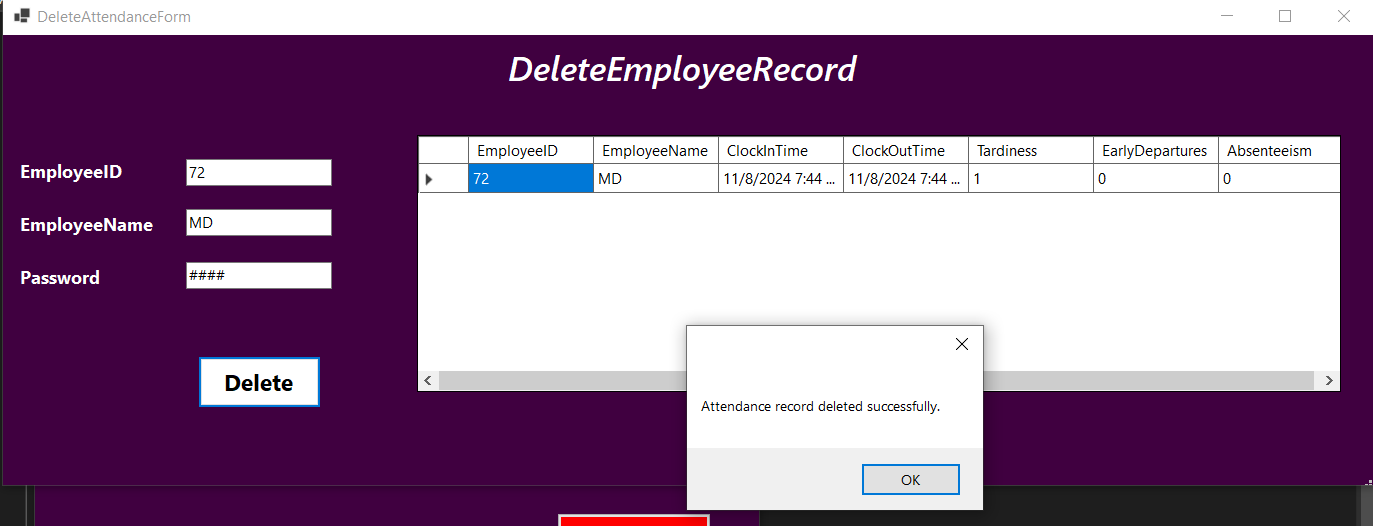
**}**

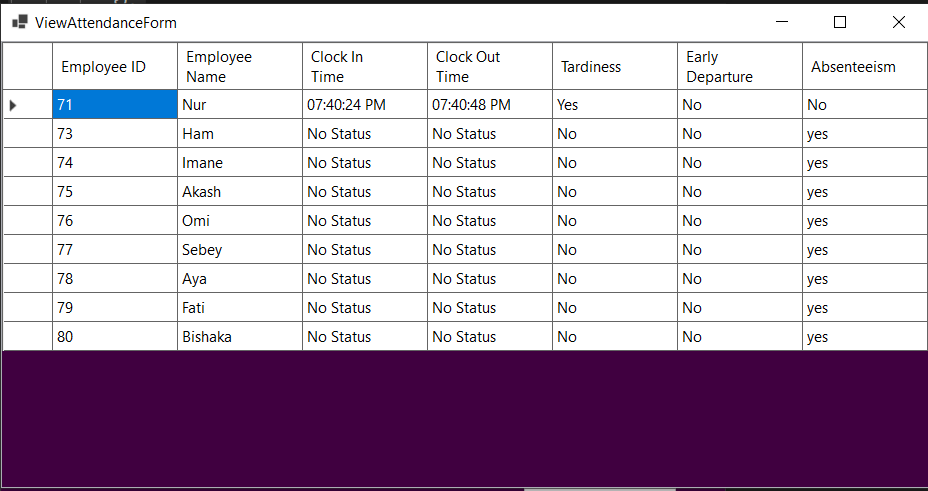
**}**

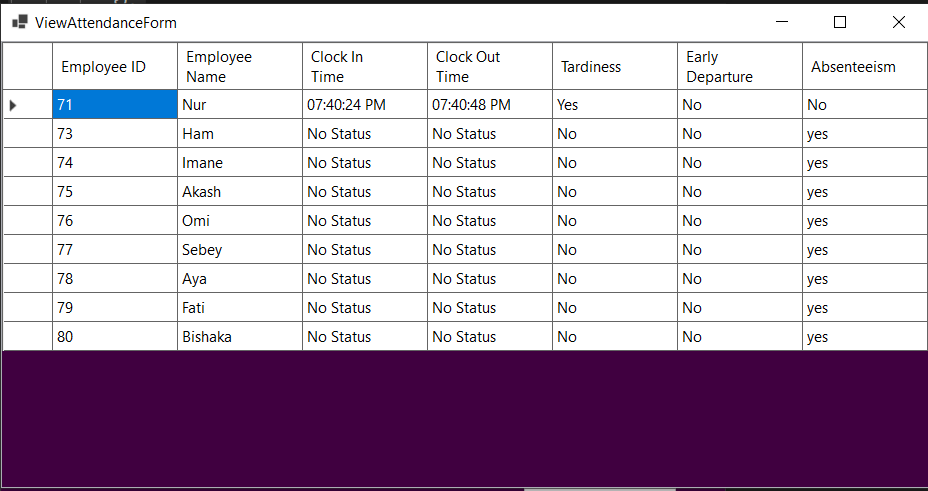
**Delete Attendance Form:**



After deleted the attendance record successfully the View Attendance Form:







**Delete Attendance Form Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**namespace EmployeeAttendancceManagementSystem**

**{**

**public partial class DeleteAttendanceForm : Form**

**{**

**private const string AdminPassword = "0000"; // Set your admin password here**

**public DeleteAttendanceForm()**

**{**

**InitializeComponent();**

**}**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**// Check if password is empty**

**if (string.IsNullOrEmpty(textBox3.Text))**

**{**

**MessageBox.Show("Please enter the password.");**

**return;**

**}**

**if (textBox3.Text != AdminPassword)**

**{**

**MessageBox.Show("Incorrect password.");**

**return;**

**}**

**var employee = EmployeeData.Employees**

**.FirstOrDefault(emp =>**

**(string.IsNullOrEmpty(textBox1.Text) || emp.EmployeeID.ToString() == textBox1.Text) &&**

**(string.IsNullOrEmpty(textBox2.Text) || emp.EmployeeName == textBox2.Text));**

**if (employee == null)**

**{**

**MessageBox.Show("Employee record not found.");**

**return;**

**}**

**EmployeeData.Employees.Remove(employee);**

**dataGridView1.DataSource = new[] { employee };**

**MessageBox.Show("Attendance record deleted successfully.");**

**}**

**}**

**}**

# **1.5 Conclusion**

## **1.5.1 Main Findings**

The Attendance Management System automates attendance tracking, searching, and record updates, supporting multiple attendance statuses and flexible search options. It meets all specified requirements, providing a streamlined, efficient, and secure tool for HR staff. The system is designed with user-friendly interfaces that facilitate easy navigation for HR personnel, minimizing the learning curve and maximizing productivity. It integrates seamlessly with existing HR software, ensuring that data flow remains uninterrupted and reliable.

## **1.5.2 Value and Alternative Insights**

This system emphasizes the advantages of automated attendance tracking, including improved accuracy, fewer manual errors, and secure data management. Additionally, it presents alternative methods for streamlining attendance management through a modular design, featuring a structured interface that allows for easy access and modification of attendance information.

## **1.5.3 Future Possibilities**

In future iterations, this system could be expanded to include:

* **Payroll Integration**: Using attendance data to calculate payroll based on hours worked, absences, and tardiness.
* **Web or Mobile Access**: Developing a web or mobile version for remote accessibility, allowing HR and employees to access attendance information from anywhere.