

EMPLOYEE MANAGEMENT SYSTEM

02

Presentation Agenda

Introduction
Program
Form1
Register
Main Form
Dashboard
Add employee
Salary
Salary Data
Employee Data

03

Introduction

Employee Management System in C#

The Employee Management System is a C# desktop application built with the .NET Framework, designed for efficient employee record management. It offers features like adding, updating, deleting, and searching employee details through a user-friendly interface, with data securely stored in a relational database. This project demonstrates practical use of OOP concepts and is a great starting point for developers interested in C#, GUI design, and database integration.

Program

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using System.Windows.Forms;
6
7  namespace EmployeeManagementSystem
8  {
9      static class Program
10     {
11         /// <summary>
12         /// The main entry point for the application.
13         /// </summary>
14         [STAThread]
15         static void Main()
16         {
17             Application.EnableVisualStyles();
18             Application.SetCompatibleTextRenderingDefault(false);
19             Application.Run(new Form1());
20         }
21     }
22 }
23
```

This file initializes the Employee Management System, sets up the environment, and launches the main form (Form1) for managing employees..

Form 1

```
• • •  
• • • 2  using System.Collections.Generic;  
• • • 3  using System.ComponentModel;  
• • • 4  using System.Data;  
• • • 5  using System.Drawing;  
• • • 6  using System.Linq;  
• • • 7  using System.Text;  
• • • 8  using System.Threading.Tasks;  
• • • 9  using System.Windows.Forms;  
• • •10  using System.Data;  
• • •11  using System.Data.SqlClient;  
• • •12  
• • •13  namespace EmployeeManagementSystem  
• • •14  {  
• • • 9 references  
• • •15  public partial class Form1 : Form  
• • •{  
• • • 17  SqlConnection connect  
• • • = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\maria\Documents\employee.mdf;Integrated Security = True;  
• • •18  
• • • 4 references  
• • •19  
• • •20 > public Form1()...  
• • •24  
• • • 1 reference  
• • •25 > private void label5_Click(object sender, EventArgs e)...  
• • •29  
• • • 1 reference  
• • •30 > private void Form1_Load(object sender, EventArgs e)...  
• • •34  
• • • 1 reference  
• • •35 > private void exit_Click(object sender, EventArgs e)...  
• • •39  
• • • 1 reference  
• • •40 > private void login_signupBtn_Click(object sender, EventArgs e)...  
• • •46  
• • • 1 reference  
• • •47 > private void login_showPass_CheckedChanged(object sender, EventArgs e)...  
• • •51  
• • • 1 reference  
• • •52 > private void login_btn_Click(object sender, EventArgs e)...
```

Handles login, validates credentials via SQL, manages UI transitions, and includes password toggle and error handling.



Form 1

Form1.cs [Design] ✎ X

Employee Management System

Register your Account

SIGNUP

Login Account

X

Username:

Password:

Show Password

LOGIN

Register Form

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Data;
11 using System.Data.SqlClient;
12 using System.Runtime.Remoting.Contexts;
13
14 namespace EmployeeManagementSystem
15 {
16     public partial class RegisterForm : Form
17     {
18         SqlConnection connect
19             = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\maria\Documents\employee.mdf;Integrated Security = True");
20         public RegisterForm()
21         {
22
23
24
25
26         }
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117 }
```

Handles user registration, checks for duplicate usernames in a SQL database, adds new users if valid, manages UI transitions, and includes password toggle and error handling.

Register Form

RegisterForm.cs [Design] X

Employee Management System

Login your Account

SIGN IN

X

Register Account

Username:

Password:

Show Password

SIGNUP

Main Form

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace EmployeeManagementSystem
12 {
13     public partial class MainForm : Form
14     {
15         public MainForm()
16         {
17             InitializeComponent();
18         }
19
20         private void exit_Click(object sender, EventArgs e)
21         {
22             Application.Exit();
23         }
24
25         private void logout_btn_Click(object sender, EventArgs e)
26         {
27             // Logic for logging out
28         }
29
30         private void dashboard_btn_Click(object sender, EventArgs e)
31         {
32             // Logic for dashboard
33         }
34
35         private void addEmployee_btn_Click(object sender, EventArgs e)
36         {
37             // Logic for adding employee
38         }
39
40         private void salary_btn_Click(object sender, EventArgs e)
41         {
42             // Logic for salary management
43         }
44
45         private void updateEmployee_btn_Click(object sender, EventArgs e)
46         {
47             // Logic for updating employee
48         }
49
50         private void deleteEmployee_btn_Click(object sender, EventArgs e)
51         {
52             // Logic for deleting employee
53         }
54
55         private void searchEmployee_btn_Click(object sender, EventArgs e)
56         {
57             // Logic for searching employee
58         }
59
60         private void updatePassword_btn_Click(object sender, EventArgs e)
61         {
62             // Logic for updating password
63         }
64
65         private void forgotPassword_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
66         {
67             // Logic for forgot password link
68         }
69
70         private void checkForDuplicates()
71         {
72             // Logic for checking for duplicates
73         }
74
75         private void validateInput()
76         {
77             // Logic for validating input
78         }
79
80         private void handleErrors()
81         {
82             // Logic for handling errors
83         }
84     }
85 }
```

Handles user registration, checks for duplicate usernames in a SQL database, adds new users if valid, manages UI transitions, and includes password toggle and error handling.

Main Form

MainForm.cs [Design] X

Employee Management System

Welcome, User

DASHBOARD ADD EMPLOYEE SALARY

Sign Out

Total Employees 3 Active Employees 3 Inactive Employees 0

Dashboard

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Data;
11 using System.Data.SqlClient;
12
13 namespace EmployeeManagementSystem
14 {
15     public partial class Dashboard : UserControl
16     {
17         SqlConnection connect
18         = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\maria\Documents\employee.mdf;Integrated Security = True; Connect
19         > 1 reference
20             public Dashboard()...
21
22         > 2 references
23             public void RefreshData()...
24
25         > 2 references
26             public void displayTE()...
27
28         > 2 references
29             public void displayAE()...
30
31         > 2 references
32             public void displayIE()...
33
34
35
36
37
38
39
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150 }
```

This C# code defines a "Dashboard" user control for an Employee Management System, displaying the total, active, and inactive employee counts from a SQL Server database with real-time data refresh functionality.

Dashboard

Dashboard.cs [Design] X



0
Total Employees



0
Active Employees



0
Inactive Employees



Add Employee

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Data;
11 using System.Data.SqlClient;
12 using System.IO;
13
14 namespace EmployeeManagementSystem
15 {
16     public partial class AddEmployee : UserControl
17     {
18         SqlConnection connect
19             = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\maria\Documents\employee.mdf;Integrated Security = True; Connect Timeout = 30");
20
21         public AddEmployee()
22         {
23
24         }
25
26         public void RefreshData()
27         {
28
29         }
30
31         public void displayEmployeeData()
32         {
33
34         }
35
36         private void addEmployee_addBtn_Click(object sender, EventArgs e)
37         {
38
39         }
40
41         private void addEmployee_importBtn_Click(object sender, EventArgs e)
42         {
43
44         }
45
46         private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
47         {
48
49         }
50
51         public void clearFields()
52         {
53
54         }
55
56         private void addEmployee_updateBtn_Click(object sender, EventArgs e)
57         {
58
59         }
60
61         private void addEmployee_clearBtn_Click(object sender, EventArgs e)
62         {
63
64         }
65
66         private void addEmployee_deleteBtn_Click(object sender, EventArgs e)
67         {
68
69         }
70     }
71 }
72 
```

Manages employees via SQL database: add, update, delete, validate, upload images, and refresh UI.

Add Employee

AddEmployee.cs [Design] X

Employee's Data

Employee ID:

Full Name:

Gender:

Phone Number:

Position:

Status:

Import

Add **Update** **Delete** **Clear**

Salary

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Data;
11 using System.Data.SqlClient;
12
13 namespace EmployeeManagementSystem
14 {
15     public partial class Salary : UserControl
16     {
17         SqlConnection connect
18         = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\maria\Documents\employee.mdf;Integrated Security = True; Connect Timeout = 30");
19
20         public Salary()
21         {
22             InitializeComponent();
23         }
24
25         private void salary_Load(object sender, EventArgs e)
26         {
27             RefreshData();
28         }
29
30         private void RefreshData()
31         {
32             try
33             {
34                 connect.Open();
35                 string query = "SELECT * FROM Employees";
36                 SqlCommand cmd = new SqlCommand(query, connect);
37                 SqlDataAdapter da = new SqlDataAdapter(cmd);
38                 DataTable dt = new DataTable();
39                 da.Fill(dt);
40                 dataGridView1.DataSource = dt;
41             }
42             catch (Exception ex)
43             {
44                 MessageBox.Show(ex.Message);
45             }
46         }
47
48         private void disableFields()
49         {
50             salary_updateBtn.Enabled = false;
51             salary_clearBtn.Enabled = false;
52             salary_nameTxt.Enabled = false;
53             salary_emailTxt.Enabled = false;
54             salary_addressTxt.Enabled = false;
55             salary_contactTxt.Enabled = false;
56             salary_salaryTxt.Enabled = false;
57             salary_hireDateTxt.Enabled = false;
58             salary_departmentTxt.Enabled = false;
59         }
60
61         private void displayEmployees()
62         {
63             try
64             {
65                 connect.Open();
66                 string query = "SELECT * FROM Employees";
67                 SqlCommand cmd = new SqlCommand(query, connect);
68                 SqlDataAdapter da = new SqlDataAdapter(cmd);
69                 DataTable dt = new DataTable();
70                 da.Fill(dt);
71                 dataGridView1.DataSource = dt;
72             }
73             catch (Exception ex)
74             {
75                 MessageBox.Show(ex.Message);
76             }
77         }
78
79         private void salary_updateBtn_Click(object sender, EventArgs e)
80         {
81             try
82             {
83                 connect.Open();
84                 string query = "UPDATE Employees SET Name = @name, Email = @email, Address = @address, Contact = @contact, Salary = @salary, HireDate = @hireDate, Department = @department WHERE Id = @id";
85                 SqlCommand cmd = new SqlCommand(query, connect);
86                 cmd.Parameters.AddWithValue("@name", salary_nameTxt.Text);
87                 cmd.Parameters.AddWithValue("@email", salary_emailTxt.Text);
88                 cmd.Parameters.AddWithValue("@address", salary_addressTxt.Text);
89                 cmd.Parameters.AddWithValue("@contact", salary_contactTxt.Text);
90                 cmd.Parameters.AddWithValue("@salary", salary_salaryTxt.Text);
91                 cmd.Parameters.AddWithValue("@hireDate", salary_hireDateTxt.Text);
92                 cmd.Parameters.AddWithValue("@department", salary_departmentTxt.Text);
93                 cmd.Parameters.AddWithValue("@id", dataGridView1.SelectedRows[0].Cells[0].Value);
94                 cmd.ExecuteNonQuery();
95                 MessageBox.Show("Employee updated successfully!");
96                 displayEmployees();
97                 disableFields();
98             }
99             catch (Exception ex)
100            {
101                MessageBox.Show(ex.Message);
102            }
103        }
104
105        private void clearFields()
106        {
107            salary_nameTxt.Clear();
108            salary_emailTxt.Clear();
109            salary_addressTxt.Clear();
110            salary_contactTxt.Clear();
111            salary_salaryTxt.Clear();
112            salary_hireDateTxt.Clear();
113            salary_departmentTxt.Clear();
114        }
115
116        private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
117        {
118            if (e.RowIndex > -1)
119            {
120                salary_nameTxt.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
121                salary_emailTxt.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
122                salary_addressTxt.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
123                salary_contactTxt.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
124                salary_salaryTxt.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();
125                salary_hireDateTxt.Text = dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString();
126                salary_departmentTxt.Text = dataGridView1.Rows[e.RowIndex].Cells[7].Value.ToString();
127            }
128        }
129
130        private void salary_clearBtn_Click(object sender, EventArgs e)
131        {
132            clearFields();
133        }
134    }
135}
```

Manages employee salary with database integration, displays data, updates records, validates input, and refreshes the UI.

Salary

Salary.cs [Design] X

Employee ID:

Full Name:

Position:

Salary:

Update **Clear**

Employees

Salary Data

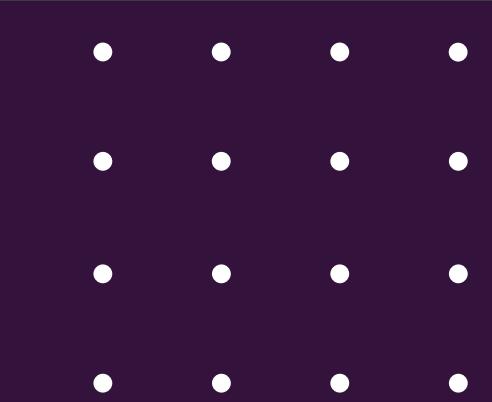
```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Data;
7  using System.Data.SqlClient;
8
9  namespace EmployeeManagementSystem
10 {
11     class SalaryData
12     {
13
14         public string EmployeeID { set; get; } // 0
15         public string Name { set; get; } // 1
16         public string Gender { set; get; } // 2
17         public string Contact { set; get; } // 3
18         public string Position { set; get; } // 4
19         public int Salary { set; get; } // 5
20
21         SqlConnection connect
22         = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\maria\Documents\employee.mdf;Integrated Security = True; Connect Timeout = 30");
23
24         >     public List<SalaryData> salaryEmployeeListData()...
25
26     }
27 }
```

Defines SalaryData to retrieve and manage employee salary info from SQL database.

Employee Data

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Data;
7  using System.Data.SqlClient;
8
9  namespace EmployeeManagementSystem
10 {
11     class EmployeeData
12     {
13
14         public int ID { set; get; } // 0
15         public string EmployeeID { set; get; } // 1
16         public string Name { set; get; } // 2
17         public string Gender { set; get; } // 3
18         public string Contact { set; get; } // 4
19         public string Position { set; get; } // 5
20         public string Image { set; get; } // 6
21         public int Salary { set; get; } // 7
22         public string Status { set; get; } // 8
23
24
25         SqlConnection connect
26             = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\maria\Documents\employee.mdf;Integrated Security = True; Connect Timeout = 30");
27
28
29         >     public List<EmployeeData> employeeListData();
30
31         >     public List<EmployeeData> salaryEmployeeListData();
32
33     }
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114 }
```

This C# class EmployeeData defines properties for employee details and includes methods to retrieve employee and salary data from a database. It uses SQL queries to fetch records while excluding



Team Members

sec 8

sec 2

sec 7

sec 10

sec 2

sec 7

sec 3

منه اللّه أَحْمَدُ بْنُ الْفَتَاحِ الْأَغَـا
آيَهُ جَمَـالُ السَّـيـدِ عَبـدـالـمـنـعـمـ
مـرـيمـ إـيـهـابـ يـحيـيـ جـمـالـ طـهـ
دـالـيـاـ طـارـقـ أـحـمـدـ تـرـكـيـ
جـيـسـكـاـ عـمـادـ فـوزـيـ لـيـبـ
مـنـهـ اللـهـ اـبـرـاهـيمـ عـبـدـ الـحـلـيمـ عـمـرـ
سـارـةـ شـرـيفـ عـلـيـ عـبـدـ الـحـمـيدـ



Thank You