Sikkim Manipal University

Directorate of Distance Education

A Project Work Report On

Human Resource Management with C#

Bachelor of Science in Information Technology (BSc IT)
6th Semester

Of

SIKKIM MANIPAL UNIVERSITY- DDE

By

Anukul Sharma

Registration No: 1302007237

Course: BSc IT

Semester: 6th Semester

LC Code: 03290
Submitted To: Niva Management & IT College

VIVA-VOCE SHEET

We have conducted the viva-voce examination of the report

SUBMITTED BY

Mr. Anukul Sharma

BSc IT 6th Semester

Roll No: 1302007237

ENTITLED

Human Resource Management System with C#

And found that the thesis to be the original work of the student and written accordingly to the prescribed format. We recommended the report to be accepted as partial fulfillment of the requirement for the degree of

Bachelor of Science in Information Technology (BSc IT)

VIVA-VOCE COMMITTEE

Head, Research Department: ………………………….
Member (Thesis Supervisor): …………………………
Member (External Expert): ……………………………

Date: - …………………

DECLARATION

I hereby declare that the project entitled "**Human Resource Management** with C#" submitted to Niva Management & IT College is a record of my original work and this project work is submitted in the partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Information Technology (BSc IT). The results embodied in this thesis has not been submitted to any other university or Institute or other degree, association ship, fellowship or any other similar titles for the award of any degree or diploma.

Signature of the Student

Mr. Anukul Sharma

Roll No: 1302007237

Course: BSc IT 6th Semester

Place: Birtamode-7, Jhapa

CERTICICATE FROM SUPERVISORS

This is to certify that the Project entitled "**Human Resource Management with C#**" submitted by **Anukul Sharma** to the Department of Computer Application, School of Science and Technology, Niva Management & IT College towards partial fulfillment of the requirements for the award of the degree of Bachelor of Science and Information Technology (BSc IT 6th Semester) is an original work carried out by him/her and meets all the requirements defined by the university to award the degree.

Signature

Name: Er. Prakash Gurung

Designation: Lecturer

(Project Supervisor)

Signature

Name:

Designation:

(External Examiner)

Signature

Name: Mr. Sandesh Malla

Designation: Principal

(Principal and Co- coordinator of Niva College)

Address: Niva Management & IT College, Birtamode, Jhapa.
Date:

ACKNOWLEDGEMENT

According to the course of study of BSc IT sixth semester determined by SMU a computer project is to be carried out for the partial fulfillment of the requirements for degree of BSc IT. Therefore as a student of this course I have developed an application "Human Resource Management System with c#".

I feel very glad for getting such an opportunity to accomplish the BSc IT sixth semester project. This project gave me insight knowledge about the practical aspect of the various stages and procedures of software development project.

This Project Report consist of explanation on the requirements and design of the new system. The requirements are started in different forms such as natural languages and diagrams to provide clear picture of each requirement.

I would like to acknowledge our sincere thanks towards our study center "Niva Management & IT College" for facilitating the completion of entire Project work.

First of all I would like to thank our supervisor, Er. Prakash Gurung sir for his valuable guidance and encouragement throughout the project. I also thank him for helping us out in difficulties by guiding us to the right direction and for the valuable suggestions which are very crucial for the success of our project and department of Computer Application for their valuable guidance and suggestions that have resulted in the successful completion of the project.

I am grateful to all those who have directly or indirectly helped me in completion of the project.

Anukul Sharma

Contents

INTRODUCTION

Human Resource Management (HRM) is the function within an organization that focuses on recruitment of, management of, and providing direction for the people who work in the organization. HRM can also be performed by line managers. HRM is the organizational function that deals with issues related to people such as compensation, hiring, performance management, organization development, safety, wellness, benefits, employee motivation, communication, administration, and training. HRM is also a strategic and comprehensive approach to managing people and the workplace culture and environment. Effective HRM enables employees to contribute effectively and productively to the overall company direction and the accomplishment of the organization's goals and objectives.

Human resource management is all about managing the Employees. Human resource management has become all that important in today's knowledge economy as it deals with processes required to organize, manage, and lead the project team. One of the key elements of human resource management is that each of the project team members should have clear assignment of their roles and responsibilities to avoid any ambiguity. Also, we should make a note that even the team working on project management tasks is part of the project team and they are responsible for Project Management functions like initiating, planning, executing, monitoring and controlling, and closing various project phases. With C# it gives us the platform to create a software which help organization in managing its employees.

The working mechanism of HRM is similar to that of coding various C# programs. Understanding how HRM works in different companies gives ideas about how we can code our program in C#. Using different syntax, functions, Windows Forms, and so on we can code Human Resource Management with C#. With a server database can be saved and kept. When any Organization uses this software they would have knowledge to keep track record of their employees and various other things.

Brief History of Human Resources Management

The history of Human Resources starts to be interesting with the evolution of the large factories. It was in the 18th century. The rapid development of new industrial approach to work changed the world dramatically. The quick and cheap production became a priority for many industries. The factories hired thousands of workers, who worked up to 16 hours a day. Soon, many entrepreneurs discovered that satisfied employees are more effective and can produce more than depressed employees. Many factories started to introduce voluntary programs for employees to increase their comfort and satisfaction. On the other hand, the government started to intervene to introduce some basic human rights and the work safety legislation.

The second rapid development of Human Resources started in the beginning of 20th century. Most organizations introduced the Personal Management. The personnel department had large responsibilities. It was dealing with issues, introducing the new law requirements. It had the responsibility for the implementation of different social and work place safety programs. Everything was focused on the productivity of employees. The regular productivity increments were the key measure for the management of employees. The significant change was introduced after the 2nd World War because the military developed many training programs for new soldiers. After the war, the training became a respected process in personnel department.During this period, the trade unions evolved. The trade unions changed the rules of the game. The employer got a strong partner to discuss with. Trade unions introduced many improvements at no significant costs for the employer. Today, trade unions are not as strong as they were used to be, but many organizations still benefit or suffer from a strong presence of trade unions in their factories.

The real HR Revolution began in 70's of the 20th century. The technology and the globalization have changed the rules of the game. Most HR Functions are running complex HRIS solutions, which make information about employees available anywhere and anytime to managers and HR Professionals. The economy of the wealthy western countries shifted towards the services economy. The quality of services became the crucial competitive advantage. HR became necessary because the structure of the workforce changed. The leadership development was the right answer.

Managers and leaders have to think global today; they have to understand to different cultural backgrounds. The corporate culture cannot be country specific; it has to reflect many nations working for the organization. This is a fantastic opportunity for Human Resources. Human Resources Management is global today. The global HR policies drive processes in different countries, but the processes produce comparable results. The employees relocate from country to country. The future of Human Resources is bright. The globalization cannot be stopped because nations collaborate. The organizations become less country specific, and they cannot identify themselves with one country. New technologies will bring other revolutions to offices. The commute working is standard today, but it will become a norm. However, the future of Human Resources will be about new networking methods and how to make employees know each other.

ABSTARCT

Proposed **"Human Resource Management System** with C#**"** is developed for to provide information of Employees. It includes modules required to successfully operate Employees without any difficulties. It has Admin functionalities to add modify and delete employee information, Trainees information who wants to enter into Firm, Leave details and Leave Application for Employees and Administrator to ask or decide whether to take leave or not. This project has included all Employee Code, Name, Designation, and all other types of information regarding Employee. It includes validation for forms to make forms distinct and avoid any difficulties.

Admin can modify individual Employee Form, Leave Form, Salary Form, and Training Form from front end and change in all forms and change reflect in all Database tables immediately.

Therefore proposed "**Human Resource Management System** with C#" has been designed to control Employee information and activities. This System can make the daily activities efficient and providing the fast response.

It included inbuilt Start Form where when we press correct number module opens to login form to enhance security features as system handles sensitive employee's information and recruitments.

Human Resource Management System Benefits

- Complete information of Employees for all the activities in the future use.
- Supports addition, update, cancellation and deletion of Employee Information to Admin forms and Leave and Loan Application for Employees for their conv.
- It is a scalable system.
- Has a clearly arranged and user-friendly interface for both Admins and Employees.
- Validation of forms in order to make data clean.
- Calculator for Employees to make their job Easier.
- All important details can be updated by front end master module only.
- Retrieval of forgotten passwords of Employees through Admins.
- Strong Security features

Propose of thesis

The thesis aims to theoretically study the use of Human Resource Management (HRM) and use in practical field using C#. Development of human resources is essential for any organization that would like to be dynamic and growth-oriented. The objective of the thesis is to plan/design, and theoretically and practically implement use of HRM in different organization that would serve as boon in the organizational context by which the employees of an organization are helped, in a continuous and planned way to:

1. Acquire or sharpen capabilities required to perform various functions associated with their present or expected future roles;
2. Develop their general capabilities as individuals and discover and exploit their own inner potentials for their own and/or organizational development purposes; and
3. Develop an organizational culture in which supervisor-subordinate relationships, teamwork and collaboration among sub-units are strong and contribute to the professional well-being, motivation and pride of employees.

Scope

The scope of Human Resource Management is very wide. :

1. Personnel aspect- This is concerned with manpower planning, recruitment, selection, placement, transfer, promotion, training and development, layoff and retrenchment, remuneration, incentives, productivity etc.

2. Welfare aspect- It deals with working conditions and amenities such as canteens, crèches, rest and lunch rooms, housing, transport, medical assistance, education, health and safety, recreation facilities, etc.

3. Industrial relations aspect- This covers union-management relations, joint consultation, collective bargaining, grievance and disciplinary procedures, settlement of disputes, etc.

Goals and Objectives

Goals

The human resources department plays an important role to the success of an organization. Depending on the organization, human resource management may also be referred to as human resources, HR or human resource development. The human resources department within an organization oversees employee relations, including determining salaries and loans, hiring staff and establishing employee performance objectives. The policies and procedures in regards to employee relations, which are coordinated by human resources management, are consistent with the overall business goals and objectives of the organization.

Objective

- To help the organization reach its goals.
- To ensure effective utilization and maximum development of human resources.
- To ensure respect for human beings. To identify and satisfy the needs of individuals.
- To ensure reconciliation of individual goals with those of the organization.
- To achieve and maintain high morale among employees.
- To provide the organization with well-trained and well-motivated employees.
- To increase to the fullest the employee’s job satisfaction and self-actualization.
- To develop and maintain a quality of work life. To be ethically and socially responsive to the needs of society.
- To develop overall personality of each employee in its multidimensional aspect.
- To enhance employee’s capabilities to perform the present job.
- To equip the employees with precision and clarity in transaction of business.
- To inculcate the sense of team spirit, team work and inter-team collaboration

Requirements

There are two requirements to be fulfilled for building this project:-

Hardware Requirements:
Processor -1 GHz, RAM- 512
Disk Space (Minimum) - 32 bit (850MB) and 64 bit (1GB).

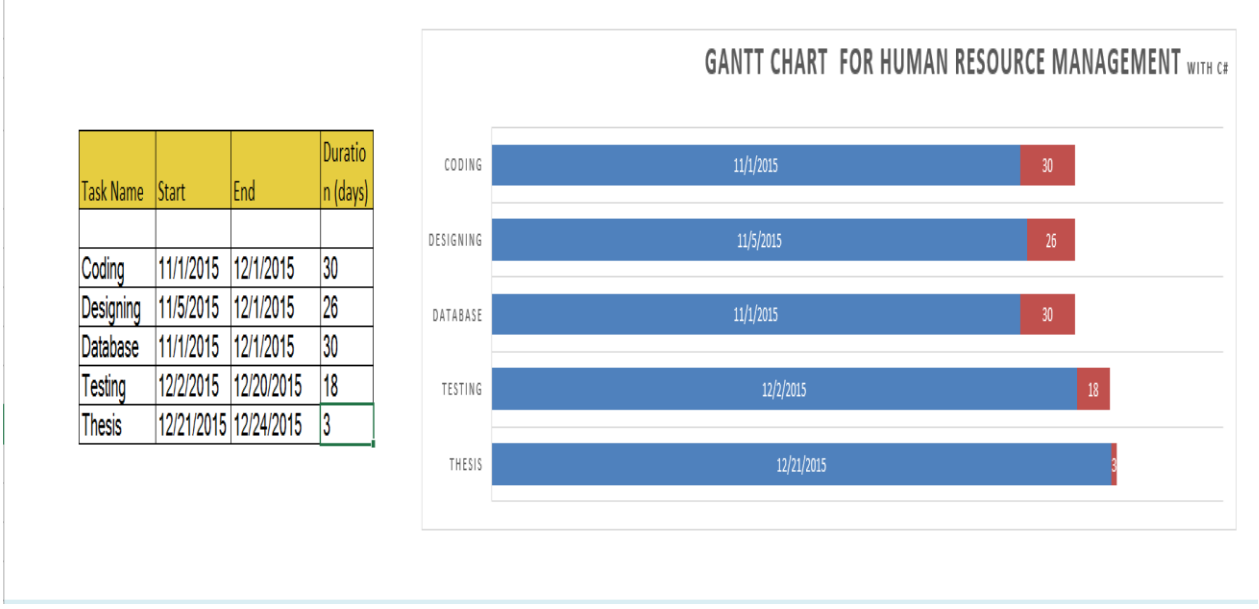
Software Requirements:
Operating Systems – Windows (for better performance higher versions of windows are used)
Software – Visual Studio and Database 2013 - Microsoft SQL server 2012.

Solution Concept

The Human Resource Management System consists of:

- **Login and Start Form**:
  Here Admins and Employees are logged in for use of this Software. First start form is opened and with correct button punched the login form appears. Username and password are given from database for better security reasons.
- **Admin or Main Form**: It includes all administrative Privileges like saving, updating, deleting, Viewing data which are connected to database. Here we can export report in Excel, Leave Report and Loan Report for viewing Employees reports too and modifying them.
- **Employee Form or Sub Main Form** – It helps Employees to apply Loan, Leave Application.
- **Validation Class**– A powerful validation is given on forms on both Admins and Employee Forms.
- **Calculator**– Calculator is Provided to Employee for their better Performance
- **Database Class** – It connects the form with Microsoft SQL Server.

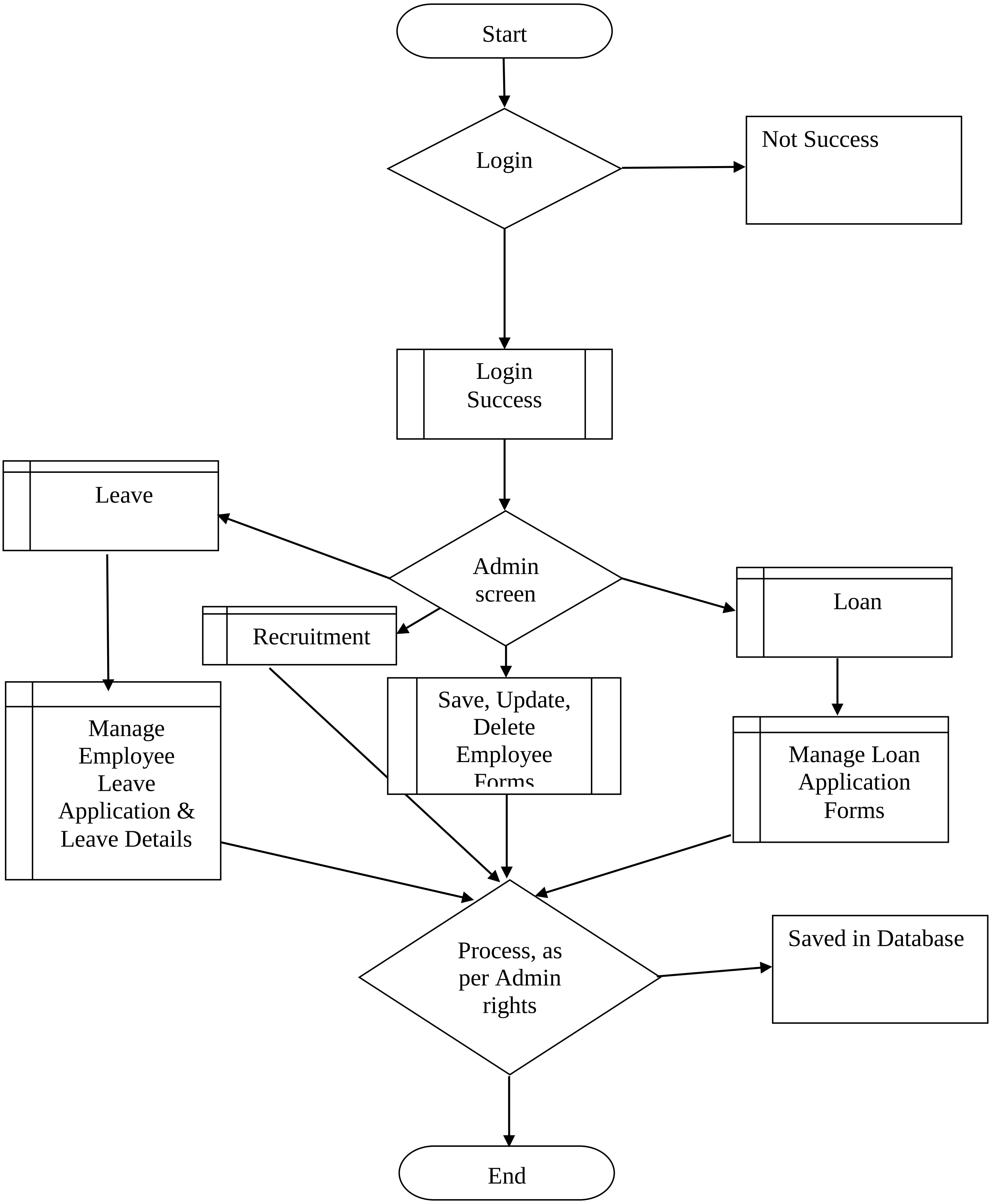
Gnatt Chart for Human Resource Management



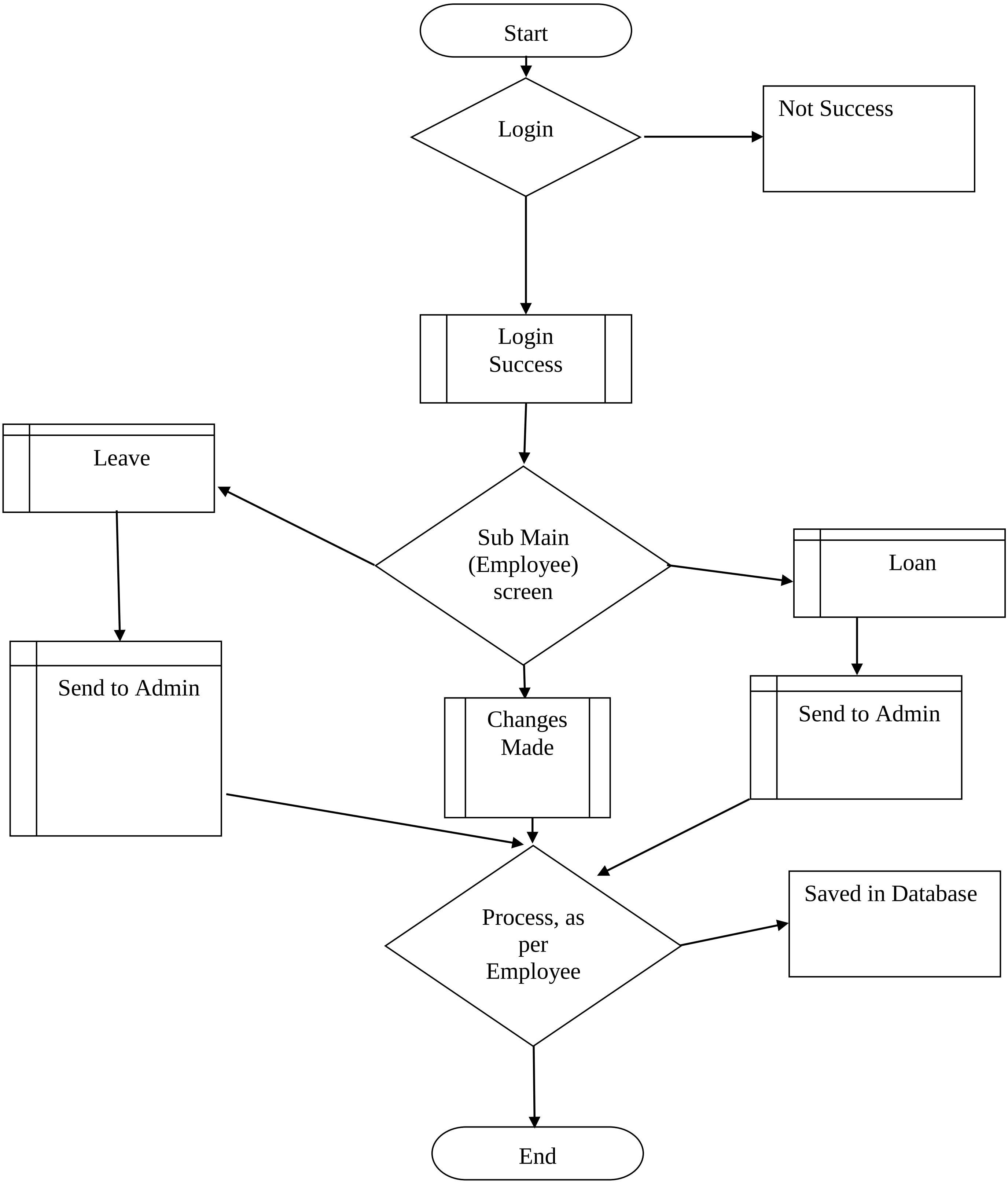
Task Name	Start	End	Duration (days)
Coding	11/1/2015	12/1/2015	30
Designing	11/5/2015	12/1/2015	26
Database	11/1/2015	12/1/2015	30
Testing	12/2/2015	12/20/2015	18
Thesis	12/21/2015	12/24/2015	3

Main and Sub Main Form Process Flow Chart

Main Form: Admin


Start
Login
Not Success
Login Success
Leave
Admin screen
Loan
Recruitment
Manage Employee Leave Application & Leave Details
Save, Update, Delete Employee Forms
Manage Loan Application Forms
Process, as per Admin rights
Saved in Database
End

Sub Main Form: Employee



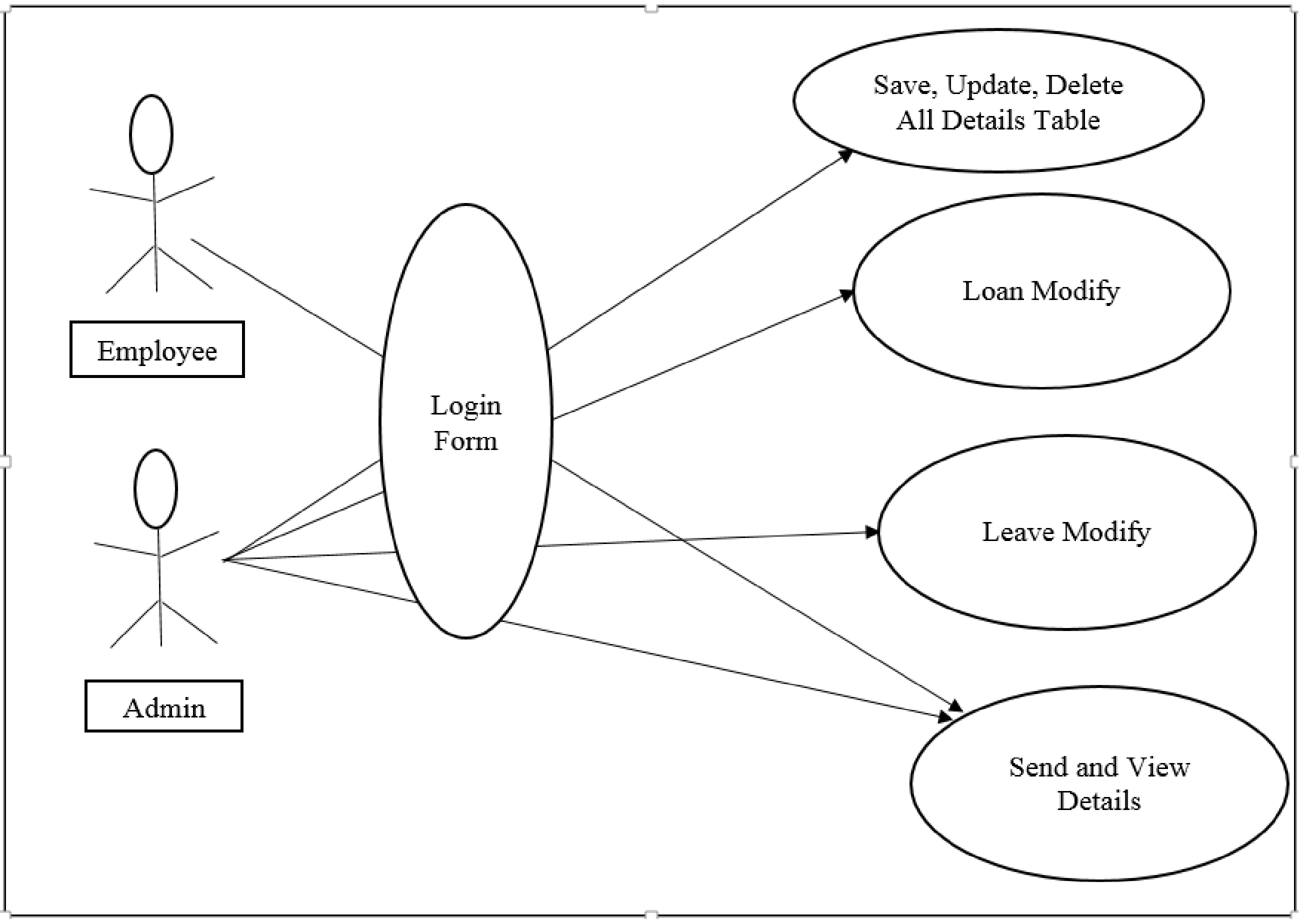
User Profiles

The following user types are expected for the **Human Resource Management with C#**:

User	**Brief Description of Use Actions**
EMPLOYEE	Minimum rights to the system, query the information can send and view.
Administrator (Admin)	Create new Employees, set and alter the role and privilege to the system users for accessing the system resource. Can delete the data. Also responsible to database backup, backend performance. And overall the system performance.

User Activity Diagram

Human Resource Management with C# is given following users' activity diagram scenarios, as represented in the following diagram:



Development Tools and Technologies

Front End: C#

Overview of C#

C# is a modern, general-purpose, object-oriented programming language developed by Microsoft and approved by European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO).

C# was developed by Anders Hejlsberg and his team during the development of .Net Framework.

C# is designed for Common Language Infrastructure (CLI), which consists of the executable code and runtime environment that allows use of various high-level languages on different computer platforms and architectures.

The following reasons make C# a widely used professional language:

- It is a modern, general-purpose programming language
- It is object oriented.
- It is component oriented.
- It is easy to learn.
- It is a structured language.
- It produces efficient programs.
- It can be compiled on a variety of computer platforms.
- It is a part of .Net Framework.

Strong Programming Features of C#

Although C# constructs closely follow traditional high-level languages, C and C++ and being an object-oriented programming language. It has strong resemblance with Java, it has numerous strong programming features that make it endearing to a number of programmers worldwide.

Following is the list of few important features of C#:

- Boolean Conditions
- Automatic Garbage Collection
- Standard Library
- Assembly Versioning
- Properties and Events
- Delegates and Events Management
- Easy-to-use Generics
- Indexers
- Conditional Compilation
- Simple Multithreading

- LINQ and Lambda Expressions
- Integration with Windows
- Conditional compilation.
- C# is pure object-oriented.
- Formalized concept of get-set methods, so the code becomes more legible.
- More clean events management (using delegates)’

Namespaces in C#

Namespace is designed for providing a way to keep one set of names separate from another. The class names declared in one namespace does not conflict with the same class names declared in another.Defining a Namespace: A namespace definition begins with the keyword namespace followed by the namespace name as follows:

```
namespace namespace_name
{
   // code declarations
}
```

Interfaces in C#

An interface is defined as a syntactical contract that all the classes inheriting the interface should follow. The interface defines the 'what' part of the syntactical contract and the deriving classes define the 'how' part of the syntactical contract. Interfaces define properties, methods, and events, which are the members of the interface. Interfaces contain only the declaration of the members. It is the responsibility of the deriving class to define the members. It often helps in providing a standard structure that the deriving classes would follow. Abstract classes to some extent serve the same purpose, however, they are mostly used when only few methods are to be declared by the base class and the deriving class implements the functionalities. Declaring Interfaces: Interfaces are declared using the interface keyword. It is similar to class declaration. Interface statements are public by default. Following is an example of an interface declaration:

```
public interface ITransactions
{
   // interface members
   void showTransaction();
   double getAmount();
}
```

Class in C#

When we define a class, you define a blueprint for a data type. This does not actually define any data, but it does define what the class name means. That is, what an object of the class consists of and what operations can be performed on that object. Objects are instances of a class. The methods and variables that constitute a class are called members of the class. Defining a Class: A class definition starts with the keyword class followed by the class name; and the class body enclosed by a pair of curly braces. Following is the general form of a class definition:

```
<access specifier> class  class_name
{
   // member variables
   <access specifier> <data type> variable1;
   <access specifier> <data type> variable2;
   ...
   <access specifier> <data type> variableN;
   // member methods
   <access specifier> <return type> method1(parameter_list)
   {
      // method body
   }
   <access specifier> <return type> method2(parameter_list)
   {
      // method body
   }
   ...
   <access specifier> <return type> methodN(parameter_list)
   {
      // method body
   }
}
```

Back End

MS SQL Server

Why MS SQL Server?

Ans: We use MS SQL Server for the following reasons: -

High performance and scalability

In many situations, Microsoft SQL Server offers better performance than an Access database. SQL Server also provides support for very large databases, up to one terabyte, which is much larger than the current limit for an Access database of two gigabytes. Finally, SQL Server works very efficiently on Microsoft Windows Servers by processing queries in parallel (using multiple native threads within a single process to handle user requests) and minimizing additional memory requirements when more users are added.

Increased availability

Microsoft SQL Server databases, can be backed, either incremental or complete, while the database is in use. Consequently, you do not have to force users to exit the database to back up data. This means your database can be running up to 24 hours a day, seven days a week.

Improved security

Microsoft SQL Server can integrate with the Windows Server operating system security to provide a single log on to the network and the database. This makes it much easier for you to administer complex security schemes. An SQL Server database on a server is also better protected because unauthorized users can't get to the database file directly but must access the server first. Inbuilt Access security has been removed in later versions. Relying on application security features in Access is never as secure as in SQL Server.

Immediate recoverability
In case of system failure (such as an operating system crash or power outage), Microsoft SQL Server has an automatic recovery mechanism that recovers a database to the last state of consistency in a matter of minutes, with no database administrator intervention. Critical applications can be up and running again right away.

Reliable distributed data and transactions
Transaction processing is a vital requirement for a system that is designed to support critical applications, such as banking and online order entry. Microsoft SQL Server supports atomic transactions with transaction logging, which guarantees that all changes performed within a transaction are either committed or rolled back.

Consistency and recoverability of a database transaction are guaranteed even in the case of system failure and in the middle of complex updates by more than one user. SQL Server treats all database changes inside a transaction as a single unit of work. By definition, either an entire transaction is completed safely and all resulting changes are reflected in the database, or the transaction is rolled back—and all changes to the database are undone.

Using a two-phase commit protocol, SQL Server can even support synchronized transactions that span more than one server—ensuring that all servers on the network are maintained in a consistent state.

Server-based processing
Microsoft designed Microsoft SQL Server from the beginning as a client/server database. Data and indexes reside on a single server computer that is often accessed over the network by many client computers. SQL Server reduces network traffic by processing database queries on the server before sending results to the client. Thus client/server application can do processing where it's done best - on the server. Applications can also use stored procedures and triggers to centralize and share application logic, business rules and policies, complex queries, and data validation and referential integrity code on the server, rather than on the client.

Development Environment

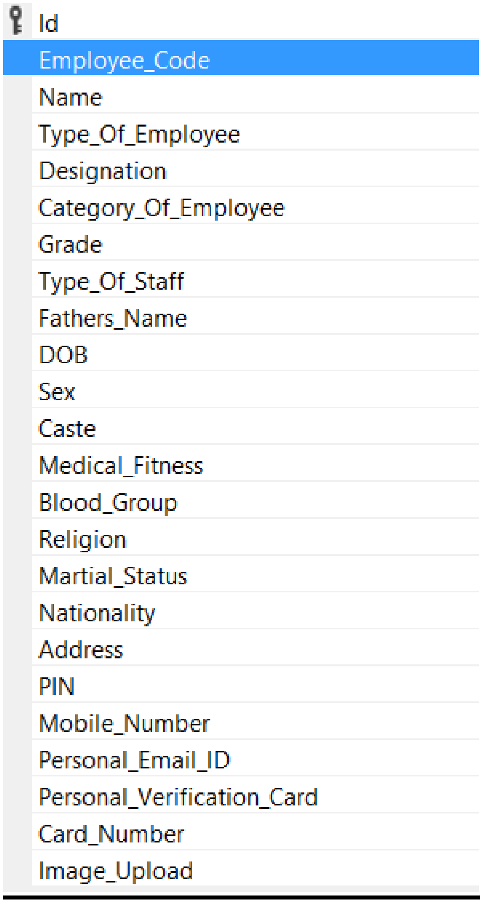
1. Visual Studio 2013.
2. Microsoft SQL Server Management Studio.

MetroFramework

Windows Modern UI for .NET Windows Forms Applications is used for better design of windows form. In this Software Application this framework is used to give effective design to it.

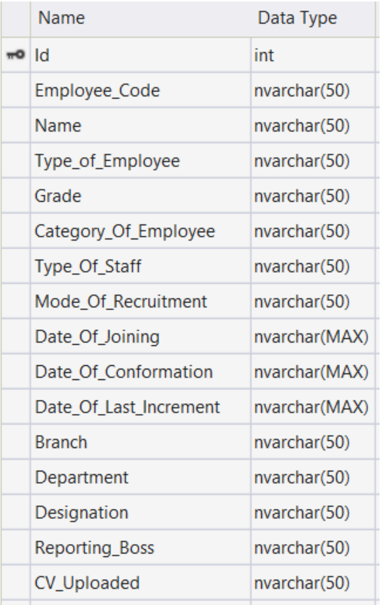
Human Resource Management System Data Dictionary

Table: Register



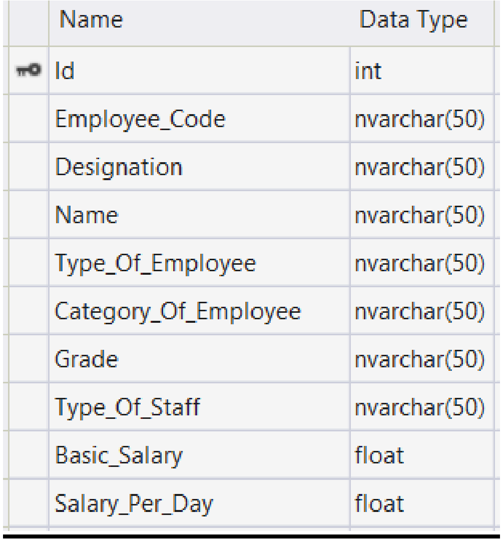
Id
Employee_Code
Name
Type_Of_Employee
Designation
Category_Of_Employee
Grade
Type_Of_Staff
Fathers_Name
DOB
Sex
Caste
Medical_Fitness
Blood_Group
Religion
Martial_Status
Nationality
Address
PIN
Mobile_Number
Personal_Email_ID
Personal_Verification_Card
Card_Number
Image_Upload

Table: Job_Details



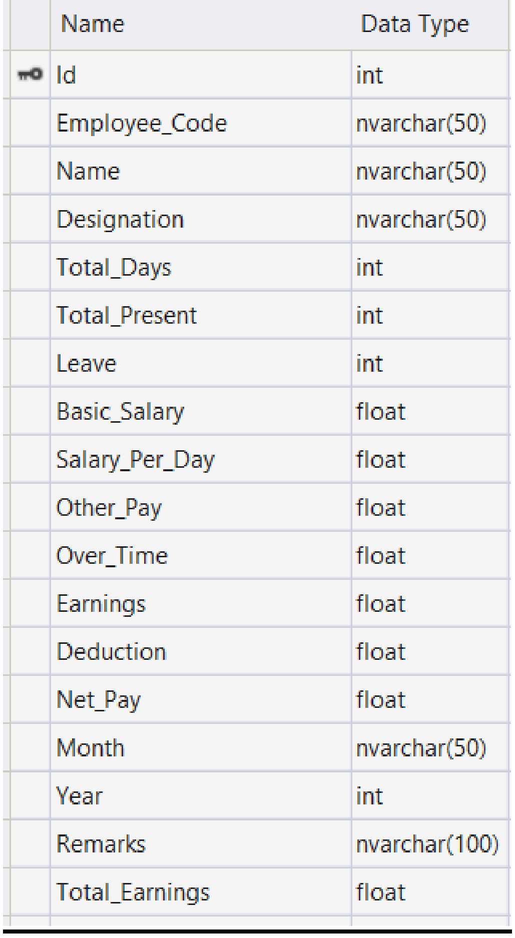
Name	Data Type
Id	int
Employee_Code	nvarchar(50)
Name	nvarchar(50)
Type_of_Employee	nvarchar(50)
Grade	nvarchar(50)
Category_Of_Employee	nvarchar(50)
Type_Of_Staff	nvarchar(50)
Mode_Of_Recruitment	nvarchar(50)
Date_Of_Joining	nvarchar(MAX)
Date_Of_Conformation	nvarchar(MAX)
Date_Of_Last_Increment	nvarchar(MAX)
Branch	nvarchar(50)
Department	nvarchar(50)
Designation	nvarchar(50)
Reporting_Boss	nvarchar(50)
CV_Uploaded	nvarchar(50)

Table: Earning_Details



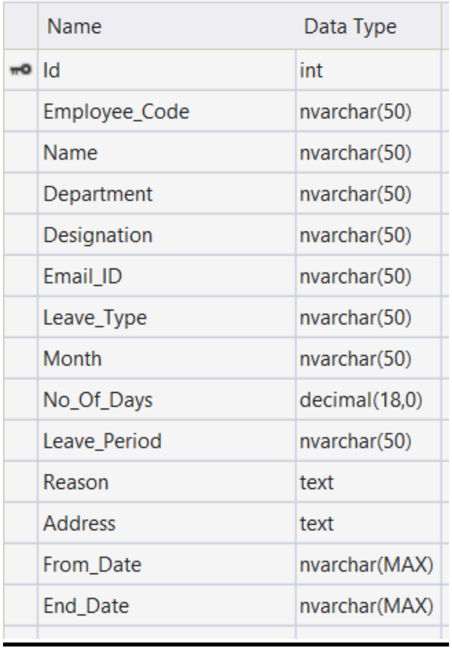
	Name	Data Type
🔑	Id	int
	Employee_Code	nvarchar(50)
	Designation	nvarchar(50)
	Name	nvarchar(50)
	Type_Of_Employee	nvarchar(50)
	Category_Of_Employee	nvarchar(50)
	Grade	nvarchar(50)
	Type_Of_Staff	nvarchar(50)
	Basic_Salary	float
	Salary_Per_Day	float

Table: Salary_Structure



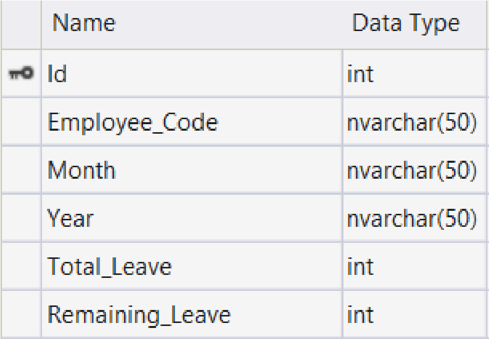
	Name	Data Type
🔑	Id	int
	Employee_Code	nvarchar(50)
	Name	nvarchar(50)
	Designation	nvarchar(50)
	Total_Days	int
	Total_Present	int
	Leave	int
	Basic_Salary	float
	Salary_Per_Day	float
	Other_Pay	float
	Over_Time	float
	Earnings	float
	Deduction	float
	Net_Pay	float
	Month	nvarchar(50)
	Year	int
	Remarks	nvarchar(100)
	Total_Earnings	float

Table: Employee_Leave



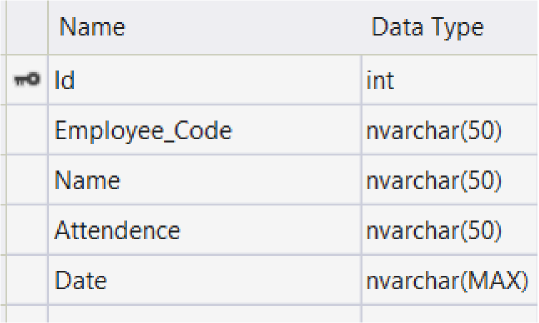
	Name	Data Type
🔑	Id	int
	Employee_Code	nvarchar(50)
	Name	nvarchar(50)
	Department	nvarchar(50)
	Designation	nvarchar(50)
	Email_ID	nvarchar(50)
	Leave_Type	nvarchar(50)
	Month	nvarchar(50)
	No_Of_Days	decimal(18,0)
	Leave_Period	nvarchar(50)
	Reason	text
	Address	text
	From_Date	nvarchar(MAX)
	End_Date	nvarchar(MAX)

Table: Monthly_Leave_Report



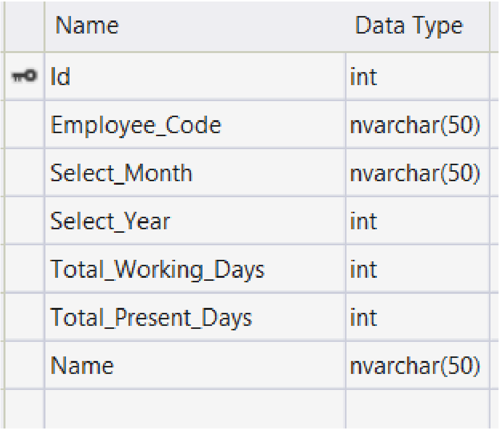
	Name	Data Type
🔑	Id	int
	Employee_Code	nvarchar(50)
	Month	nvarchar(50)
	Year	nvarchar(50)
	Total_Leave	int
	Remaining_Leave	int

Table: Daily_ Attendence_Report



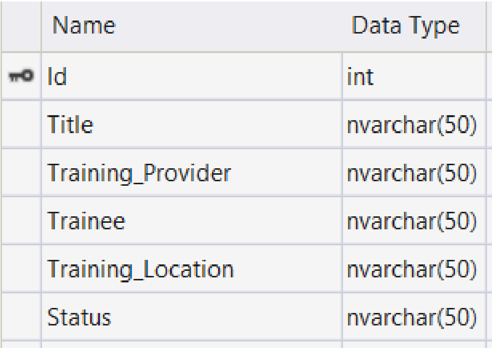
	Name	Data Type
🔑	Id	int
	Employee_Code	nvarchar(50)
	Name	nvarchar(50)
	Attendence	nvarchar(50)
	Date	nvarchar(MAX)

Table: Employee_Monthly_Attendence



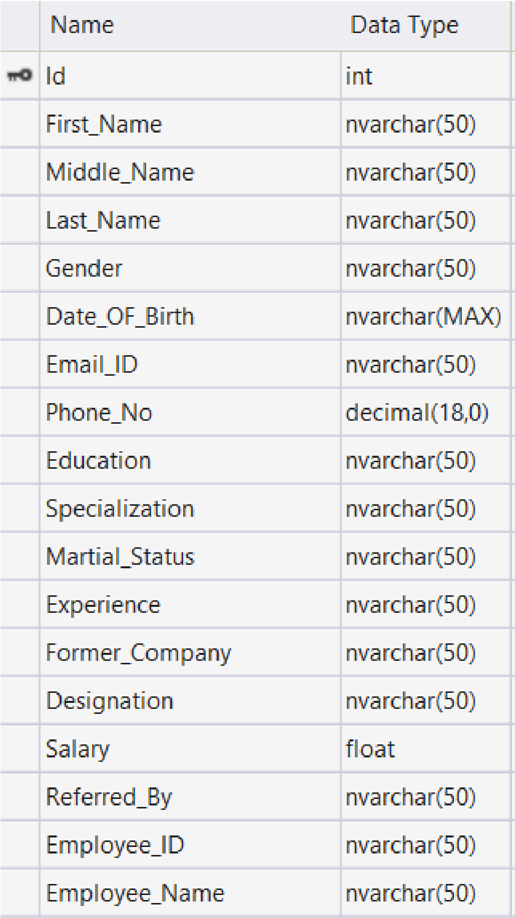
	Name	Data Type
🔑	Id	int
	Employee_Code	nvarchar(50)
	Select_Month	nvarchar(50)
	Select_Year	int
	Total_Working_Days	int
	Total_Present_Days	int
	Name	nvarchar(50)

Table:Employee_Training



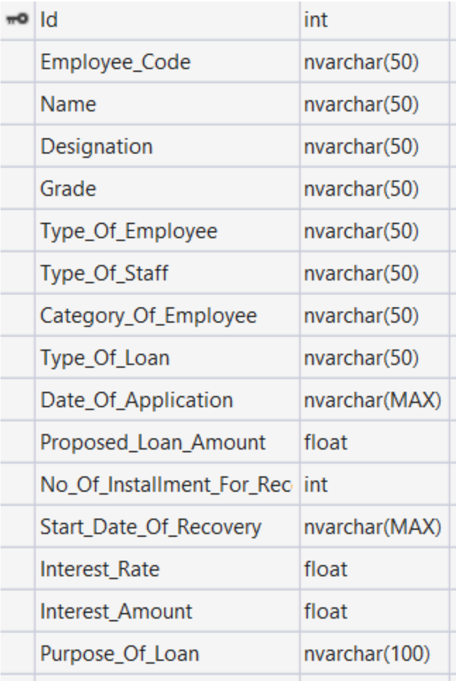
	Name	Data Type
🔑	Id	int
	Title	nvarchar(50)
	Training_Provider	nvarchar(50)
	Trainee	nvarchar(50)
	Training_Location	nvarchar(50)
	Status	nvarchar(50)

Table: Recruitment



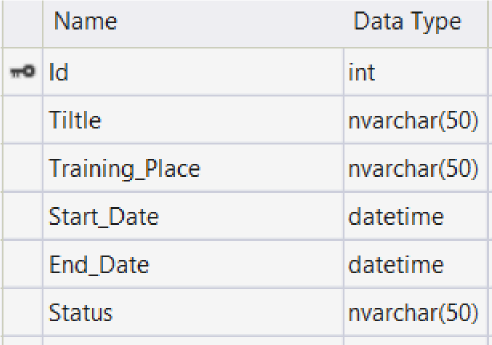
	Name	Data Type
🔑	Id	int
	First_Name	nvarchar(50)
	Middle_Name	nvarchar(50)
	Last_Name	nvarchar(50)
	Gender	nvarchar(50)
	Date_OF_Birth	nvarchar(MAX)
	Email_ID	nvarchar(50)
	Phone_No	decimal(18,0)
	Education	nvarchar(50)
	Specialization	nvarchar(50)
	Martial_Status	nvarchar(50)
	Experience	nvarchar(50)
	Former_Company	nvarchar(50)
	Designation	nvarchar(50)
	Salary	float
	Referred_By	nvarchar(50)
	Employee_ID	nvarchar(50)
	Employee_Name	nvarchar(50)

Table: Employee_Loan



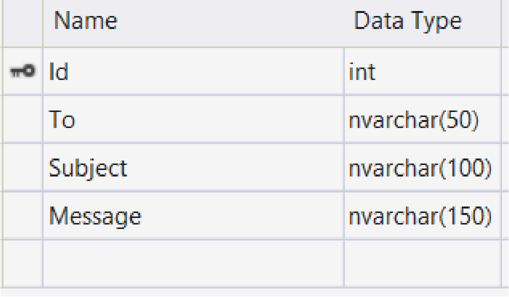
🔑	Id	int
	Employee_Code	nvarchar(50)
	Name	nvarchar(50)
	Designation	nvarchar(50)
	Grade	nvarchar(50)
	Type_Of_Employee	nvarchar(50)
	Type_Of_Staff	nvarchar(50)
	Category_Of_Employee	nvarchar(50)
	Type_Of_Loan	nvarchar(50)
	Date_Of_Application	nvarchar(MAX)
	Proposed_Loan_Amount	float
	No_Of_Installment_For_Rec	int
	Start_Date_Of_Recovery	nvarchar(MAX)
	Interest_Rate	float
	Interest_Amount	float
	Purpose_Of_Loan	nvarchar(100)

Table: Training_Events



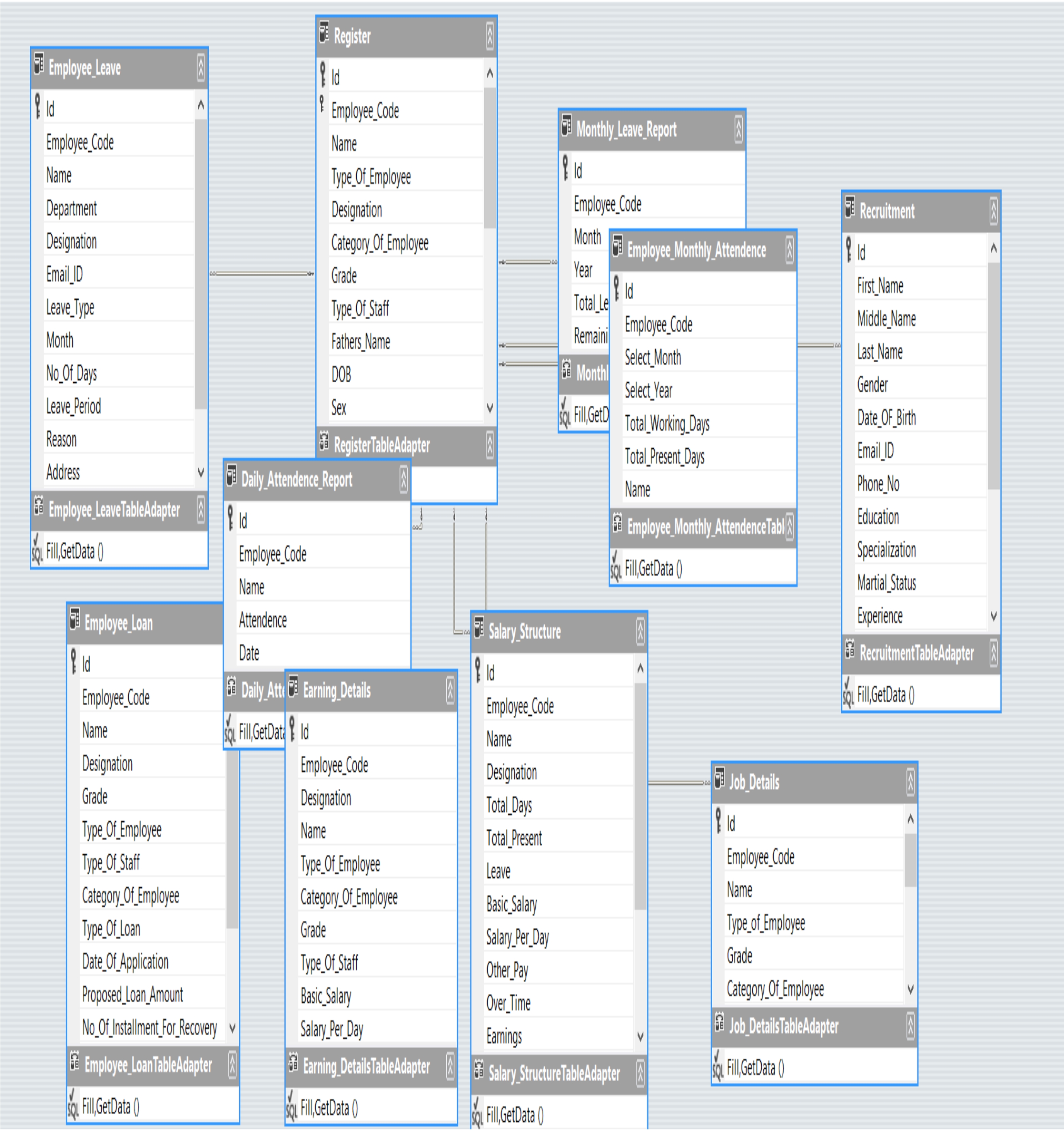
	Name	Data Type
🔑	Id	int
	Tiltle	nvarchar(50)
	Training_Place	nvarchar(50)
	Start_Date	datetime
	End_Date	datetime
	Status	nvarchar(50)

Table: Announcement



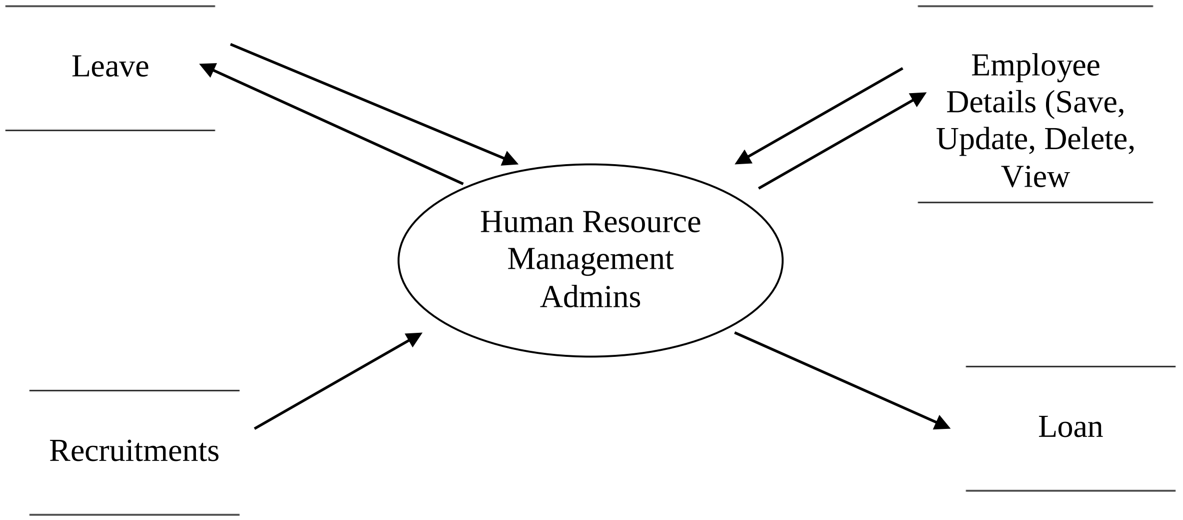
	Name	Data Type
🔑	Id	int
	To	nvarchar(50)
	Subject	nvarchar(100)
	Message	nvarchar(150)

Human Resource Management System with C# – Data Schema ER Diagram

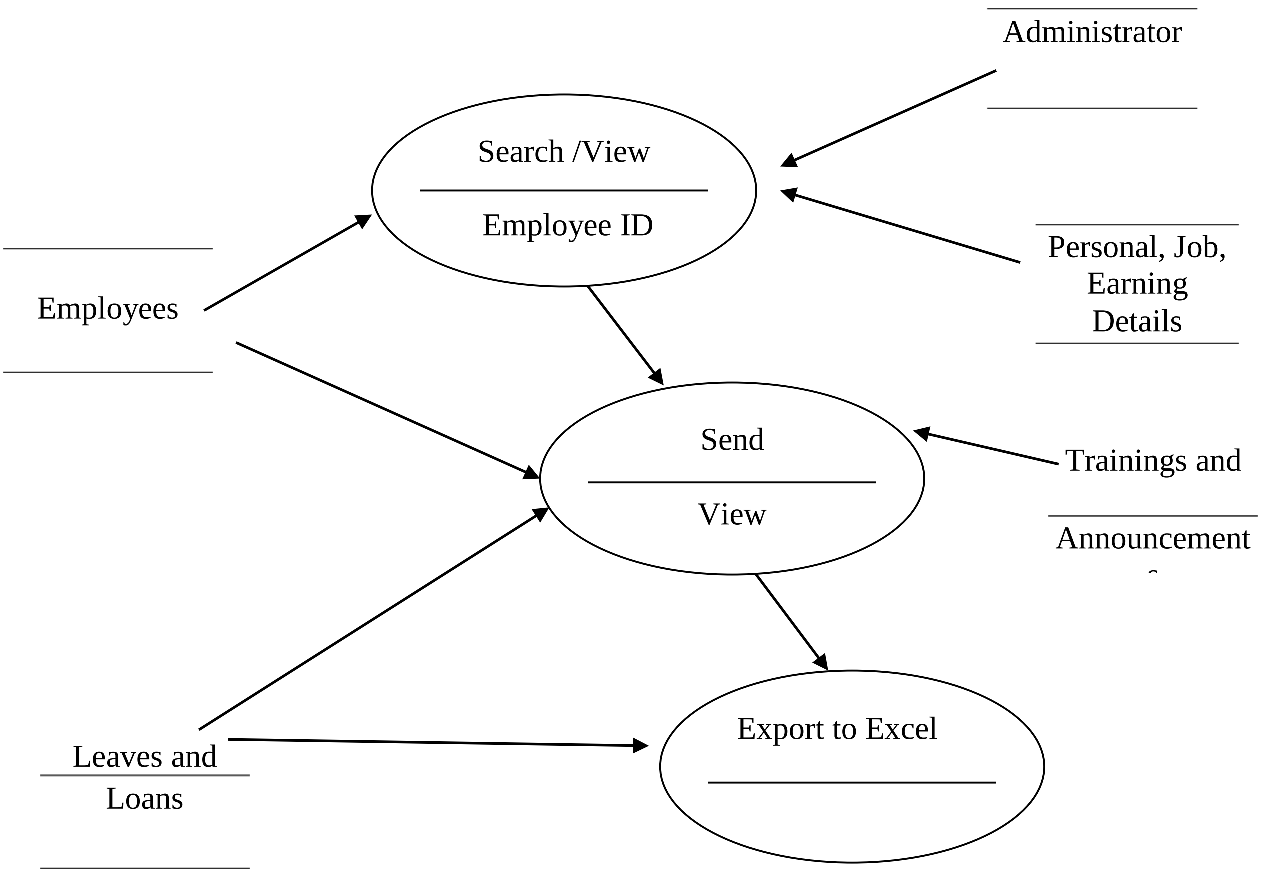


Human Resource Management with C# DFD (Data Flow Diagram)

Top Level DFD

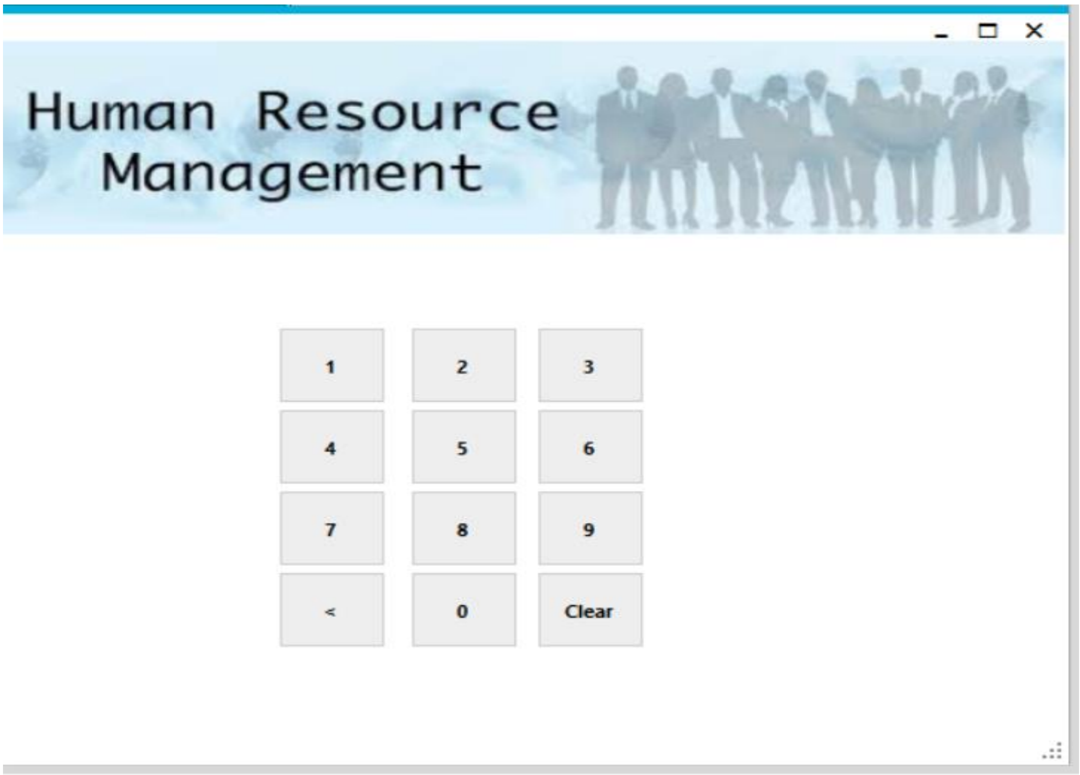

Leave
Employee Details (Save, Update, Delete, View
Human Resource Management Admins
Recruitments
Loan

1st Level DFD



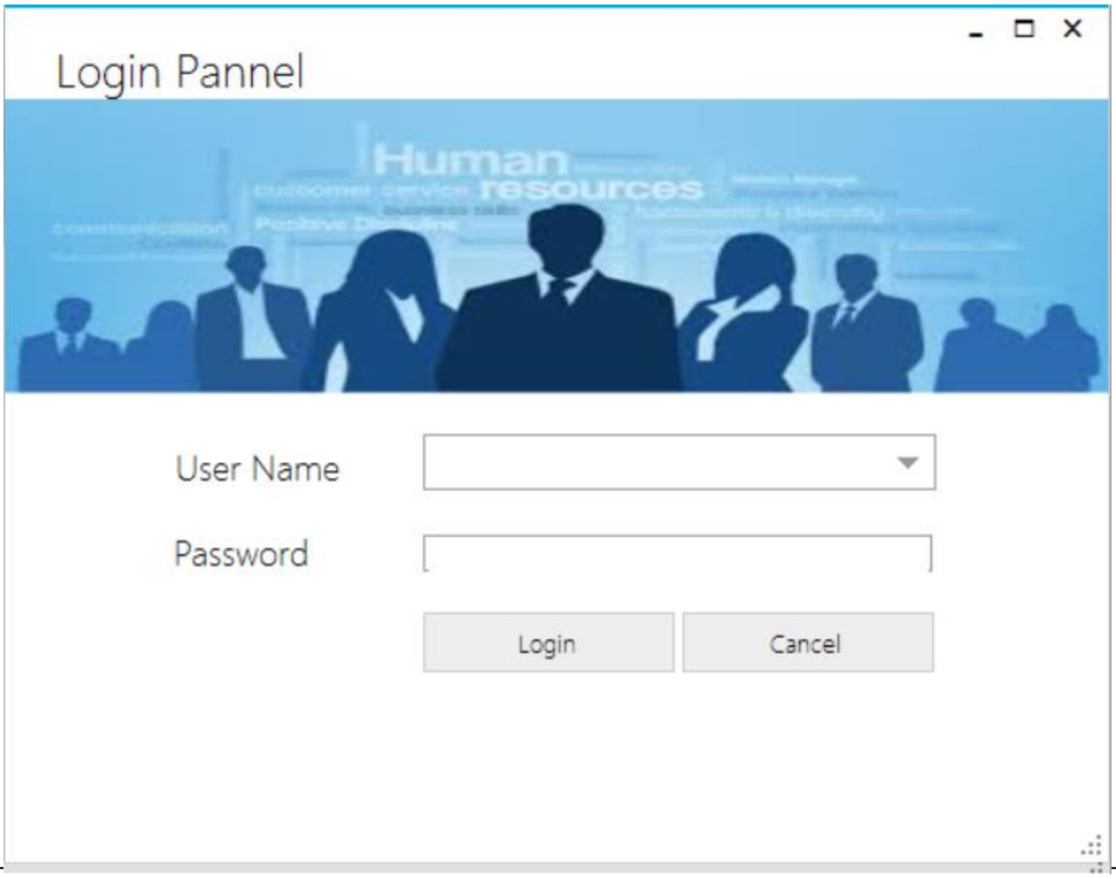
Coding and Interface

Human Resource Management with C#: Login



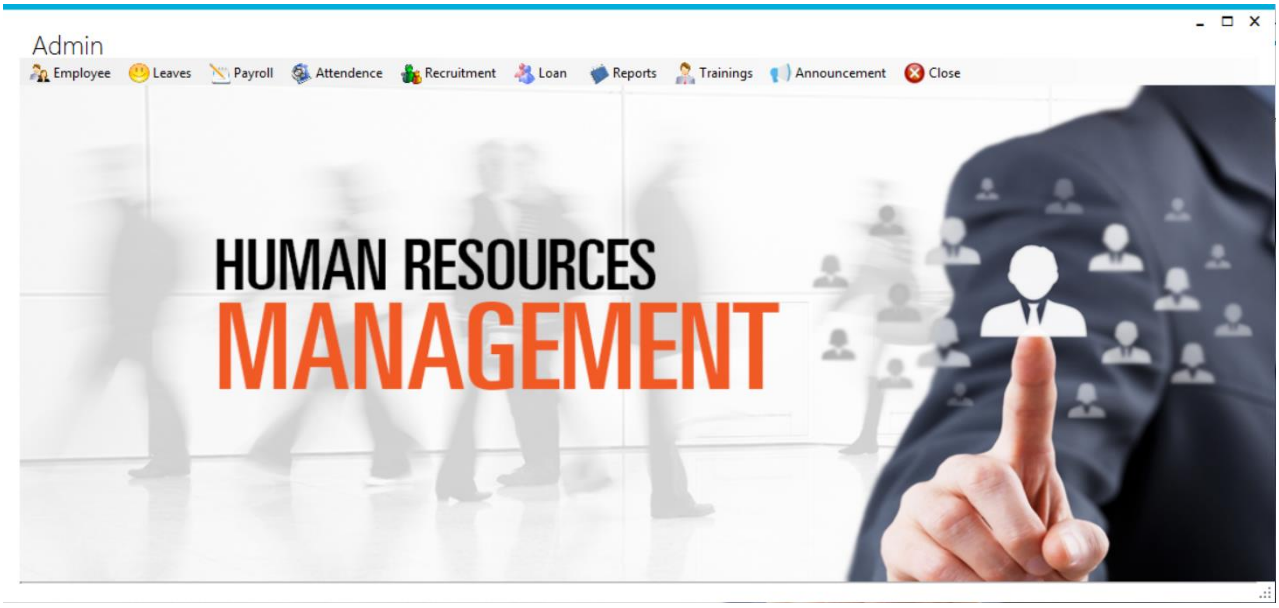
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MetroFramework.Forms;

namespace Human_Resourse_Management
{
    public partial class Login : MetroForm
    {
        string pressedNumber;
        public Login()
        {
            InitializeComponent();
        }

        private void checkLogin()
        {
            if (this.pressedNumber == "123456")
            {
                new Start().ShowDialog();
                this.Hide();
            }
        }


        private void metroButton1_Click(object sender, EventArgs e)
        {

        }

        private void Form2_Load(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button1.Text;
            checkLogin();
            // check for login

        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button2.Text;
            // check for login
            checkLogin();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button3.Text;
            checkLogin();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button4.Text;
            checkLogin();
        }

        private void button5_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button5.Text;
            checkLogin();
        }

        private void button6_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button6.Text;
            checkLogin();
        }

        private void button7_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button7.Text;
            checkLogin();
        }

        private void button8_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button8.Text;
            checkLogin();
        }

        private void button9_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button9.Text;
            checkLogin();
        }

        private void button0_Click(object sender, EventArgs e)
        {
            this.pressedNumber += button0.Text;
            checkLogin();
        }

        private void buttonClear_Click(object sender, EventArgs e)
        {
            this.pressedNumber = "";

        }

        private void metroLink1_Click(object sender, EventArgs e)
        {

        }

    }
}
```

Human Resource Management with C#:



```
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;
using MetroFramework.Forms;


namespace Human_Resourse_Management
{
    public partial class Start : MetroForm
    {
        public Start()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void mtBtlogin_Click(object sender, EventArgs e)
        {
            if (txtUsername.Text == "Admin")
            {
                if (txtPassword.Text == "pass@w0rd1")
                {
                    new Main().Show();
                    this.Hide();
                }
                else
                {
                    MessageBox.Show("Error: Invalid Password or
Username !");
                }
                return;
            }

                if (txtUsername.Text == "Employee")
                {
                    if (txtPassword.Text == "123456")
                    {
                        new SubMain().Show();
                        this.Hide();
                    }
                }
                else
                {
                    MessageBox.Show("Error: Invalid Password or
Username !");
                }


        }
        private void mtBtcancel_Click(object sender, EventArgs e)
        {
            this.Close();
            this.Hide();
        }

        private void txtUsername_SelectedIndexChanged(object sender, 
EventArgs e)
        {

        }
    }
}
```

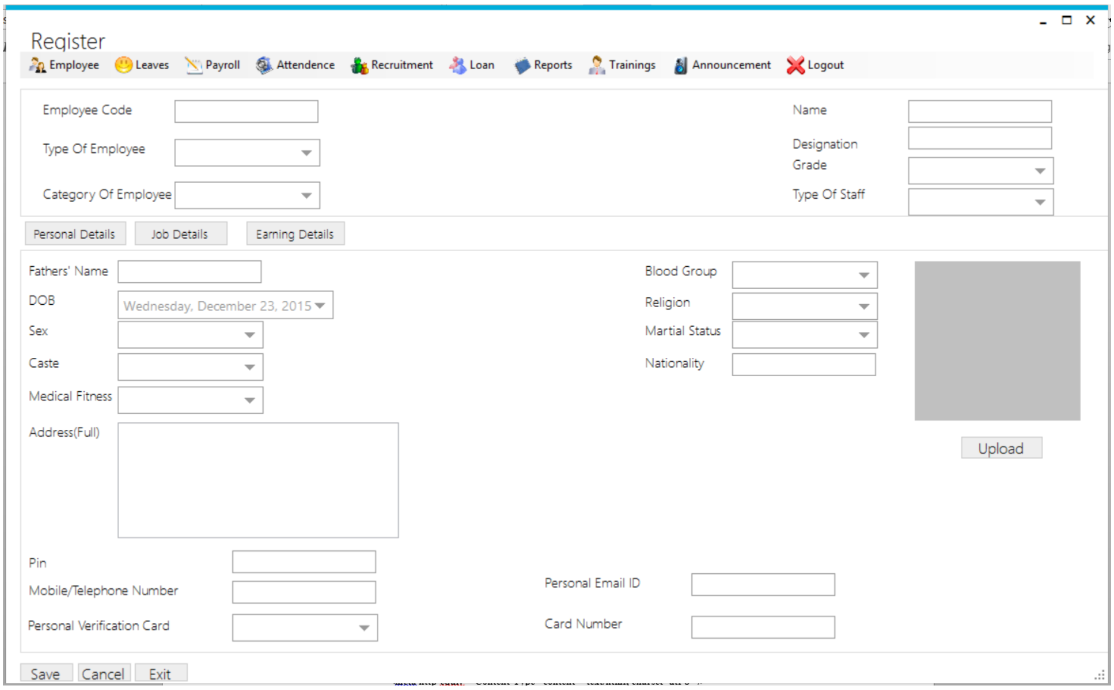
Human Resource Management with C#: Main Form (Admin)



```
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;
using MetroFramework.Forms;

namespace Human_Resourse_Management
{
    public partial class Main : MetroForm
    {
        public Main()
        {
            InitializeComponent();
        }
        private void Main_Load(object sender, EventArgs e)
        {

        }

        private void editEmployeeToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Edit_Personal_Details().ShowDialog();

        }

        private void trainingsToolStripMenuItem_Click(object sender,
EventArgs e)
        {

        }

        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }

        private void addNewEmployeeToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Register().ShowDialog();

        }

        private void editEmployeeToolStripMenuItem1_Click(object
sender, EventArgs e)
        {
            new Edit_Job_Details().ShowDialog();

        }

        private void
editEmployeeEarningDetailToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            new Edit_Earning_Details().ShowDialog();

        }

        private void employeeLeavesToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new
Human_Resourse_Management.Admin.Employee_Leave().ShowDialog();
        }

        private void leaveApplicationToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Leave_Application().ShowDialog();

        }
        private void salaryStructureToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Salary_Structure().ShowDialog();

        }

        private void employeeTrainingsToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Employee_Training().ShowDialog();

        }

        private void trainingEventToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Employee_Training().ShowDialog();

        }

        private void announcementToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Announcement().ShowDialog();

        }

        private void employeeLoanToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new
Human_Resourse_Management.Employee.Employee_Loan().ShowDialog();

        }

        private void
monthlyAttendenceReportToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            new
Human_Resourse_Management.Admin.Monthly_Attendence_Report().ShowDialo
g();

        }

        private void
dailyAttendenceReportToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            new
Human_Resourse_Management.Admin.Daily_Attendence_Report().ShowDialog(
);

        }

        private void attaindainceToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new
Human_Resourse_Management.Admin.Employees_Monthly_Attendance().ShowDi
alog();
```

```
        }

        private void leaveDetailToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new
Human_Resourse_Management.Admin.Monthly_Leave_Report().ShowDialog();

        }

        private void registerToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            new
Human_Resourse_Management.Admin.Recruitment().ShowDialog();

        }

        private void loanApplicationToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new
Human_Resourse_Management.Admin.Loan_Application().ShowDialog();
        }

        private void recruitmentDetailToolStripMenuItem_Click(object
sender, EventArgs e)
        {

        }

        private void Main_FormClosing(object sender,
FormClosingEventArgs e)
        {
            DialogResult dialog = MessageBox.Show("Do you Really want
to Exit the Program", "**********Human Resource Management
System*********", MessageBoxButtons.OKCancel);
            if (dialog == DialogResult.OK)
            {
                Application.Exit();
            }
            else if (dialog == DialogResult.Cancel)
            {
                e.Cancel = true;
            }
        }

        private void closeToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }


    }
}
```

Human Resource Management with C#: Register (Admin)



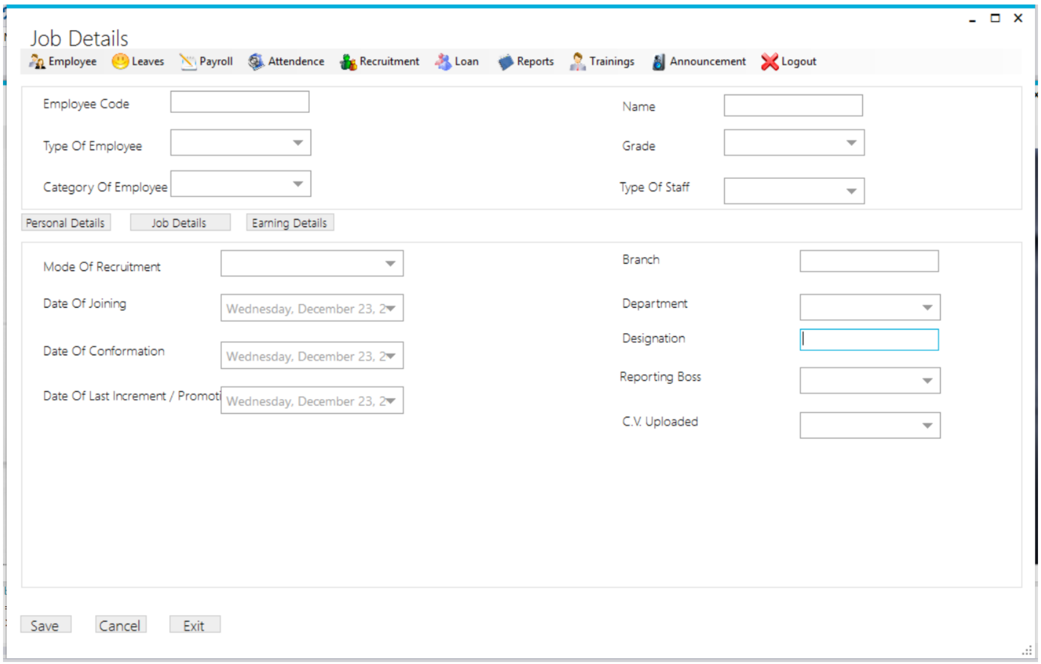
DOB Wednesday, December 23, 2015

```
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;
using MetroFramework.Forms;
using Human_Resourse_Management.Admin;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    public partial class Register : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();

        public Register()
        {
            InitializeComponent();
        }
        public Register(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *  FROM  Register
where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }

        }

        private void Register_Load(object sender, EventArgs e)
        {
            this.txtEmpCode.Focus();

        }
        private void editEmployeeToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            new Edit_Personal_Details().Show();
            this.Hide();
        }

        private void addNewEmployeeToolStripMenuItem_Click(object sender,
EventArgs e)
        {
        }

        private void editEmployeeToolStripMenuItem2_Click(object sender, EventArgs
e)
        {
            new Edit_Earning_Details().Show();

        }

        private void editEmployeeToolStripMenuItem1_Click(object sender, EventArgs
e)
        {
            new Edit_Job_Details().Show();

        }
```

```
private void logoutToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}

private void metroTextBox3_Click(object sender, EventArgs e)
{

}

private void btnUpload_Click(object sender, EventArgs e)
{
    OpenFileDialog opf = new OpenFileDialog();
    opf.Filter = "Choose Image(*.jpg; *.png; *.gif)|*.jpg; *.png; *.gif";
    if (opf.ShowDialog() == DialogResult.OK)
    {
        pictureBox1.Image = Image.FromFile(opf.FileName);
    }
}

private void groupBox2_Enter(object sender, EventArgs e)
{

}

private void btnExit_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnCancel_Click(object sender, EventArgs e)
{
    foreach (Control c in groupBox1.Controls)
    {
        txtEmpCode.Text = "";
        txtName.Text = "";
        comboTypeOfEmployee.Text = "";
        txtDesignation.Text = "";
        comboCategoryOfEmployee.Text= "";
        ComboGrade.Text = "";
        comboTypeOfStaff.Text = "";
    }
    foreach (Control b in groupBox2.Controls)
    {
       txtFathersName.Text = "";
       DateTime_DOB.Text = "";
       comboSex.Text = "";
       comboCaste.Text = "";
       comboMedicalFitness.Text = "";
       comboBloodGroup.Text = "";
       comboReligion.Text = "";
```

```
                comboMartialStatus.Text = "";
                txtNationality.Text = "";
                txtAddress.Text = "";
                txtPin.Text = "";
                txtPhoneNo.Text = "";
                txtEmailID.Text = "";
                comboPersonalVerificationCard.Text = "";
                txtCardNumber.Text = "";
            }
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtName.Text != "")
            {
                string dt = DateTime_DOB.Text;
                SqlConnection con = new SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO Register
(Employee_Code,Name,Type_Of_Employee,Designation,Category_Of_Employee,Gr
ade,Type_Of_Staff,Fathers_Name,DOB,Sex,Caste,Medical_Fitness,Blood_Group,Rel
igion,Martial_Status,Nationality,Address,PIN,Mobile_Number,Personal_Email_ID,P
ersonal_Verification_Card,Card_Number,Image_Upload) VALUES('" +
txtEmpCode.Text + "','" + txtName.Text + "','" + comboTypeOfEmployee.Text + "','"
+ txtDesignation.Text + "','" + comboCategoryOfEmployee.Text + "','" +
ComboGrade.Text + "','" + comboTypeOfStaff.Text + "','" + txtFathersName.Text +
"','" + DateTime_DOB.Text + "','" + comboSex.Text + "','" + comboCaste.Text + "','"
+ comboMedicalFitness.Text + "','" + comboBloodGroup.Text + "','" +
comboReligion.Text + "','" + comboMartialStatus.Text + "','" + txtNationality.Text +
"','" + txtAddress.Text + "','" + txtPin.Text + "','" + txtPhoneNo.Text + "','" +
txtEmailID.Text + "','" + comboPersonalVerificationCard.Text + "','" +
txtCardNumber.Text + "','" + pictureBox1.Image + "')", con);
                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                }
                catch (SqlException ex) { throw ex; }
                finally { con.Close(); }
                MessageBox.Show("Congratulations!!!, Your Data is Saved");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }

        }

        private void groupBox1_Enter(object sender, EventArgs e)
        {
        }

        private void comboTypeOfEmployee_SelectedIndexChanged(object sender,
EventArgs e)
        {

        }



        private void txtName_KeyPress(object sender, KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtEmailID_KeyPress(object sender, KeyPressEventArgs e)
        {
            Validation.checkEmail(this.txtEmailID.Text, e);
        }

        private void txtEmpCode_KeyPress(object sender, KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }

        private void txtNationality_KeyPress(object sender, KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtFathersName_KeyPress(object sender, KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtPin_KeyPress(object sender, KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtPhoneNo_KeyPress(object sender, KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtCardNumber_KeyPress(object sender, KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }

        private void txtDesignation_KeyPress(object sender, KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtEmailID_Click(object sender, EventArgs e)
        {

        }

        private void btnPersonalDetails_Click(object sender, EventArgs e)
        {
            new Register().Show();
            this.Hide();
        }

        private void btnJobDetails_Click(object sender, EventArgs e)
        {
            new Job_Details().Show();
            this.Hide();
        }

        private void btnEarningDetails_Click(object sender, EventArgs e)
        {
            new Earning_Details().Show();
            this.Hide();
        }
    }
}
```

Human Resource Management with C#: Job Details (Admin)



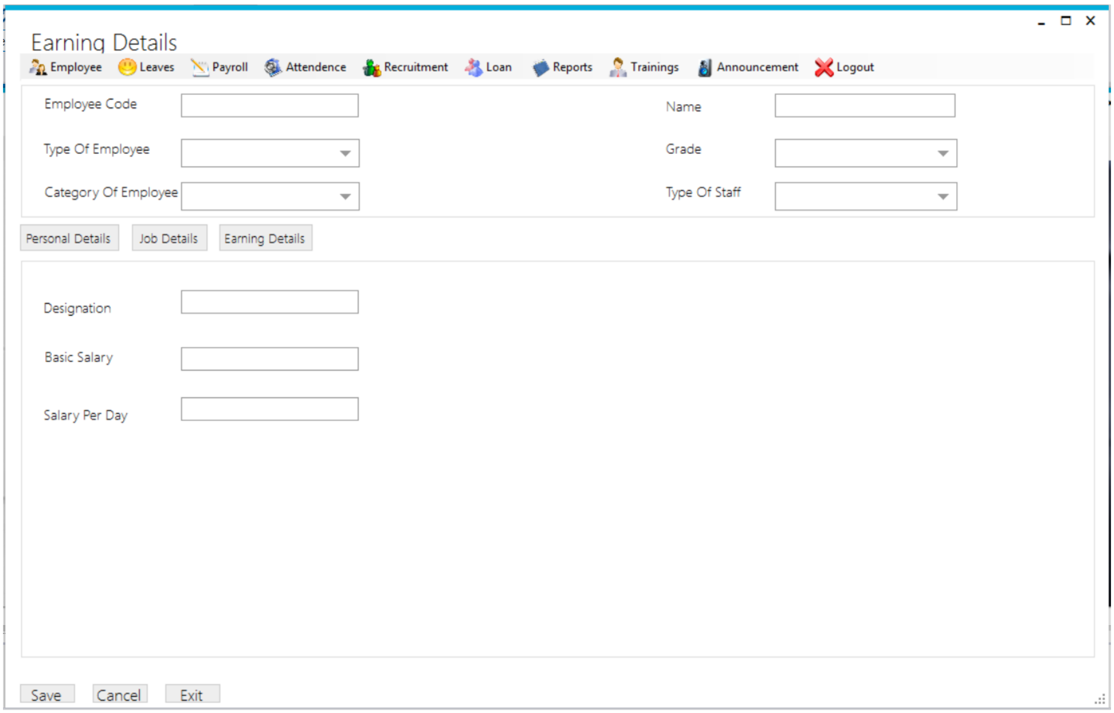
Date Of Joining: Wednesday, December 23, 2

Date Of Conformation: Wednesday, December 23, 2

Date Of Last Increment / Promoti: Wednesday, December 23, 2

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    public partial class Job_Details : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Job_Details()
        {
            InitializeComponent();
        }

        public Job_Details(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Register where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text =    dt.Rows[0][1].ToString();
                }
            }

        }
        private void Job_Detailcs_Load(object sender, EventArgs e)
        {

        }

        private void metroButton1_Click(object sender, EventArgs e)
        {
            new Register().Show();
            this.Hide();
        }

        private void metroButton2_Click(object sender, EventArgs e)
        {
            new Job_Details().Show();
            this.Hide();
        }

        private void groupBox2_Enter(object sender, EventArgs e)
        {
        }

        private void metroButton3_Click(object sender, EventArgs e)
        {
            new Earning_Details().Show();
            this.Hide();

        }

        private void addNewEmployeeToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Register().Show();
            this.Hide();
        }

        private void editEmployeeToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Edit_Personal_Details().Show();
            this.Hide();
        }

        private void editEmployeeToolStripMenuItem1_Click(object
sender, EventArgs e)
        {

        }

        private void editEmployeeToolStripMenuItem2_Click(object
sender, EventArgs e)
        {
            new Edit_Earning_Details().Show();
            this.Hide();
        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            foreach (Control c in groupBox1.Controls)
            {
                txtEmpCode.Text = "";
                txtName.Text = "";
                ComboGrade.Text = "";
                comboCategoryOfEmployee.Text = "";
                comboTypeOfEmployee.Text = "";
                comboTypeOfStaff.Text = "";

            }
            foreach (Control b in groupBox2.Controls)
            {
                comboModeOfRecruitment.Text = "";
                comboReportingBoss.Text = "";
                txtBranch.Text = "";
                comboCVUploaded.Text = "";
                txtDesignation.Text = "";
                comboDepartment.Text = "";
                DateTime_DateOfJoining.Text = "";
                DateTime_DateOfConformation.Text = "";
                DateTime_DateOfLastIncrement.Text = "";
            }
        }
        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtName.Text != "")
            {
                string dt1 = DateTime_DateOfJoining.Text;
                string dt2 = DateTime_DateOfConformation.Text;
                string dt3 = DateTime_DateOfLastIncrement.Text;
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Job_Details
(Employee_Code,Name,Type_Of_Employee,Grade,Category_Of_Employee,Type_
Of_Staff,Mode_Of_Recruitment,Date_Of_Joining,Date_Of_Conformation,Dat
e_Of_Last_Increment,Branch,Department,Designation,Reporting_Boss,CV_U
ploaded) VALUES('" + txtEmpCode.Text + "','" + txtName.Text + "','" +
comboTypeOfEmployee.Text + "','" + ComboGrade.Text + "','" +
comboCategoryOfEmployee.Text + "','" + comboTypeOfStaff.Text + "','"
+ comboModeOfRecruitment.Text + "','" + DateTime_DateOfJoining.Text +
"','" + DateTime_DateOfConformation.Text + "','" +
DateTime_DateOfLastIncrement.Text + "','" + txtBranch.Text + "','" +
comboDepartment.Text + "','" + txtDesignation.Text + "','" +
comboReportingBoss.Text + "','" + comboCVUploaded.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Saved");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void groupBox1_Enter(object sender, EventArgs e)
        {

        }

        private void txtEmpCode_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }

        private void txtName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtBranch_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }
        private void txtDesignation_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }
    }
}
```

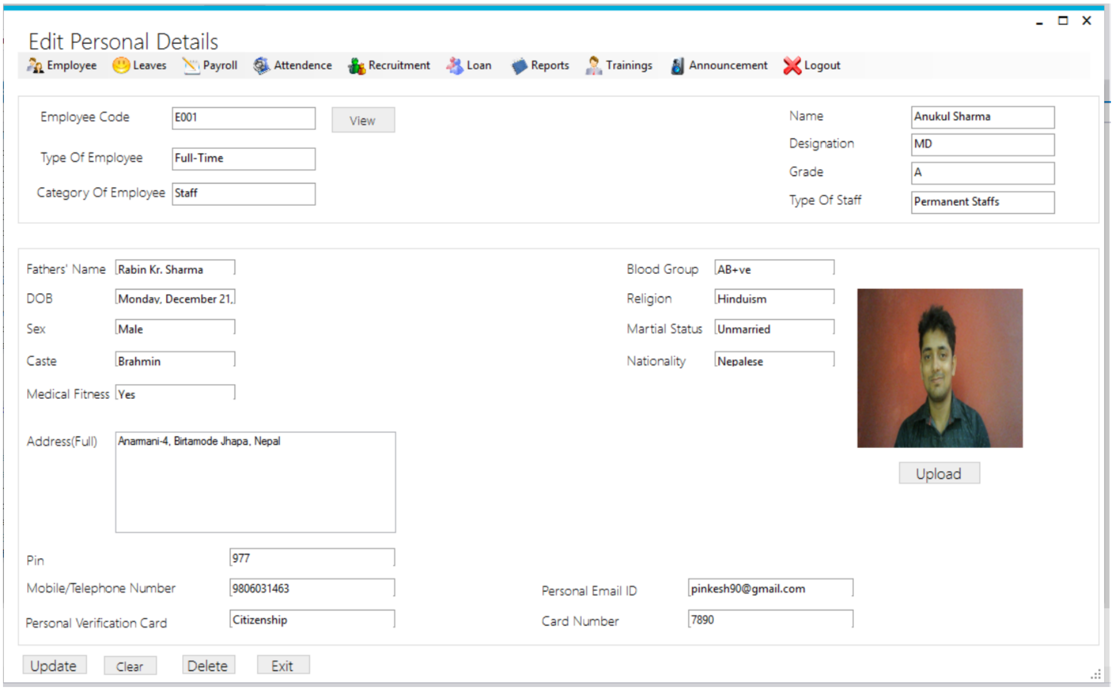
Human Resource Management with C#: Earning Details (Admin)



```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;
```

```
namespace Human_Resourse_Management
{
    public partial class Earning_Details : MetroForm

    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Earning_Details()
        {
            InitializeComponent();
        }
          public Earning_Details(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Earning_Details where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }

        }

        private void Earning_Details_Load(object sender, EventArgs e)
        {

        }

        private void metroButton1_Click(object sender, EventArgs e)
        {
            new Register().Show();
            this.Hide();

        }

        private void metroButton2_Click(object sender, EventArgs e)
        {
            new Job_Details().Show();
            this.Hide();

        }

        private void metroButton3_Click(object sender, EventArgs e)
        {
            new Earning_Details().Show();
            this.Hide();

        }

        private void metroButton6_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            foreach (Control c in groupBox1.Controls)
            {
                txtEmpCode.Text = "";
                txtName.Text = "";
                comboCategoryOfEmployee.Text = "";
                ComboGrade.Text = "";
                comboTypeOfEmployee.Text = "";
                comboTypeOfStaff.Text = "";
            }
            foreach (Control b in groupBox2.Controls)
            {
                txtDesignation.Text = "";
                txtBasicSalary.Text = "";
                txtSalaryPerDay.Text = "";
            }
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtName.Text != "")
            {
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Earning_Details
(Employee_Code,Designation,Name,Type_Of_Employee,Category_Of_Employee
,Grade,Type_OF_Staff,Basic_Salary,Salary_Per_Day) VALUES('" +
txtEmpCode.Text + "','" + txtDesignation.Text + "','" + txtName.Text
+ "','" + comboTypeOfEmployee.Text + "','" +
comboCategoryOfEmployee.Text + "','" + ComboGrade.Text + "','" +
comboTypeOfStaff.Text + "','" + txtBasicSalary.Text + "','" +
txtSalaryPerDay.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Saved");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void txtEmpCode_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }

        private void txtName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtBasicSalary_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtSalaryPerDay_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtDesignation_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }
    }
}
```

Human Resource Management with C#: Edit Personal Details (Admin)



Employee Code	E001	Name	Anukul Sharma
Type Of Employee	Full-Time	Designation	MD
Category Of Employee	Staff	Grade	A
		Type Of Staff	Permanent Staffs

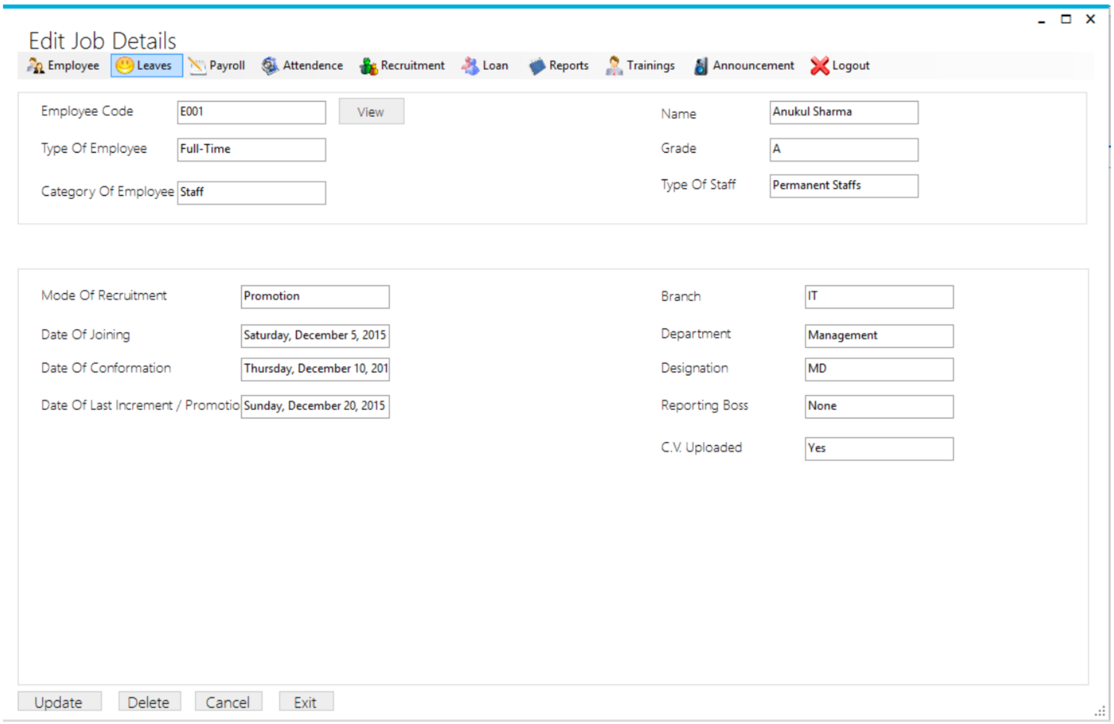
Fathers' Name	Rabin Kr. Sharma	Blood Group	AB+ve
DOB	Monday, December 21,	Religion	Hinduism
Sex	Male	Martial Status	Unmarried
Caste	Brahmin	Nationality	Nepalese
Medical Fitness	Yes		
Address(Full)	Anamani-4, Birtamode Jhapa, Nepal		
Pin	977		
Mobile/Telephone Number	9806031463	Personal Email ID	pinkesh90@gmail.com
Personal Verification Card	Citizenship	Card Number	7890

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    public partial class Edit_Personal_Details : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();

        public Edit_Personal_Details()
        {
            InitializeComponent();
        }
        public Edit_Personal_Details(int ID)
        {
            InitializeComponent();
            btnUpdate.Text = "Update";
            btnDelete.Text = "Delete";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Register where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }

        }

        private void Edit_Personal_Details_Load(object sender,
EventArgs e)
        {

        }

        private void addNewEmployeeToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Register().Show();
            this.Hide();
        }

        private void editEmployeeToolStripMenuItem_Click(object
sender, EventArgs e)
        {

        }

        private void editEmployeeToolStripMenuItem1_Click(object
sender, EventArgs e)
        {
            new Edit_Job_Details().Show();
            this.Hide();
        }

        private void editEmployeeToolStripMenuItem2_Click(object
sender, EventArgs e)
```

```
        {
            new Edit_Earning_Details().Show();
            this.Hide();
        }

        private void metroButton8_Click(object sender, EventArgs e)
        {

        }

        private void metroButton1_Click(object sender, EventArgs e)
        {

        }

        private void btnEdit_Click(object sender, EventArgs e)
        {

        }

        private void btnSave_Click(object sender, EventArgs e)
        {

        }

        private void btnUpload_Click(object sender, EventArgs e)
        {
            OpenFileDialog opf = new OpenFileDialog();
            opf.Filter = "Choose Image(*.jpg; *.png; *.gif)|*.jpg;
*.png; *.gif";
            if (opf.ShowDialog() == DialogResult.OK)
            {
                pictureBox1.Image = Image.FromFile(opf.FileName);
            }
        }

        private void btnCancel_Click(object sender, EventArgs e)
        {

        }

        private void btnView_Click(object sender, EventArgs e)
        {
            if (txtEmpCode.Text != "")
            {
                try
                {
                    SqlConnection con = new
SqlConnection(db.connectionString);
                    string sql = "SELECT *  FROM   Register WHERE
Employee_Code='" + this.txtEmpCode.Text + "'";
                    SqlCommand cmd1 = new SqlCommand(sql, con);
                    SqlDataAdapter adp = new SqlDataAdapter();
                    adp.SelectCommand = cmd1;
                    DataTable dt = new DataTable();
                    adp.Fill(dt);
                    // now get the values
                    txtName.Text = dt.Rows[0][2].ToString();
                    comboTypeOfEmployee.Text =
dt.Rows[0][3].ToString();
                    txtDesignation.Text = dt.Rows[0][4].ToString();
                    comboCategoryOfEmployee.Text =
dt.Rows[0][5].ToString();
                    ComboGrade.Text = dt.Rows[0][6].ToString();
                    comboTypeOfStaff.Text = dt.Rows[0][7].ToString();
                    txtFathersName.Text = dt.Rows[0][8].ToString();
                    DateTime_DOB.Text = dt.Rows[0][9].ToString();
                    comboSex.Text = dt.Rows[0][10].ToString();
                    comboCaste.Text = dt.Rows[0][11].ToString();
                    comboMedicalFitness.Text =
dt.Rows[0][12].ToString();
                    comboBloodGroup.Text = dt.Rows[0][13].ToString();
                    comboReligion.Text = dt.Rows[0][14].ToString();
                    comboMartialStatus.Text =
dt.Rows[0][15].ToString();
                    txtNationality.Text = dt.Rows[0][16].ToString();
                    txtAddress.Text = dt.Rows[0][17].ToString();
                    txtPin.Text = dt.Rows[0][18].ToString();
                    txtPhoneNo.Text = dt.Rows[0][19].ToString();
                    txtEmailID.Text = dt.Rows[0][20].ToString();
                    comboPersonalVerificationCard.Text =
dt.Rows[0][21].ToString();
                    txtCardNumber.Text = dt.Rows[0][22].ToString();




                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.ToString());
                }
            }
            }
        private void btnUpdate_Click(object sender, EventArgs e)
        {
            if (txtEmpCode.Text != "")
            {
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("UPDATE Register SET
Name='" + this.txtName.Text + "',Type_Of_Employee='" +
this.comboTypeOfEmployee.Text + "',Designation='" +
this.txtDesignation.Text + "',Category_Of_Employee='" +
this.comboCategoryOfEmployee.Text + "',Grade='" +
this.ComboGrade.Text + "',Type_Of_Staff='" +
this.comboTypeOfStaff.Text + "',Fathers_Name='" +
this.txtFathersName.Text + "',DOB='" + this.DateTime_DOB.Text +
"',Sex='" + this.comboSex.Text + "',Caste='" + this.comboCaste.Text +
"',Medical_Fitness='" + this.comboMedicalFitness.Text +
"',Blood_Group='" + this.comboBloodGroup.Text + "',Religion='" +
this.comboReligion.Text + "',Martial_Status='" +
this.comboMartialStatus.Text + "',Nationality='" +
this.txtNationality.Text + "',Address='" + this.txtAddress.Text +
"',PIN='" + this.txtPin.Text + "',Mobile_Number='" +
this.txtPhoneNo.Text + "',Personal_Email_ID='" + this.txtEmailID.Text
+ "',Personal_Verification_Card='" +
this.comboPersonalVerificationCard.Text + "',Card_Number='" +
this.txtCardNumber.Text + "',Image_Upload='" + this.pictureBox1.Image
+ "' WHERE Employee_Code='" + this.txtEmpCode.Text + "'", con);
                 try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                }
                catch (SqlException ex) { throw ex; }
                finally { con.Close(); }
                MessageBox.Show("Data is Updated");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }
        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }

        private void txtName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void btnClear_Click(object sender, EventArgs e)
        {
            foreach (Control c in groupBox1.Controls)
            {
                txtEmpCode.Text = "";
                txtName.Text = "";
                comboTypeOfEmployee.Text = "";
                txtDesignation.Text = "";
                comboCategoryOfEmployee.Text= "";
                ComboGrade.Text = "";
                comboTypeOfStaff.Text = "";
            }
            foreach (Control b in groupBox2.Controls)
            {
               txtFathersName.Text = "";
               DateTime_DOB.Text = "";
               comboSex.Text = "";
               comboCaste.Text = "";
               comboMedicalFitness.Text = "";
               comboBloodGroup.Text = "";
               comboReligion.Text = "";
               comboMartialStatus.Text = "";
               txtNationality.Text = "";
               txtAddress.Text = "";
               txtPin.Text = "";
               txtPhoneNo.Text = "";
               txtEmailID.Text = "";
               comboPersonalVerificationCard.Text = "";
               txtCardNumber.Text = "";
            }
        }

        private void btnDelete_Click(object sender, EventArgs e)
        {

            SqlConnection con = new
SqlConnection(db.connectionString);
```

```
            SqlCommand cmd = new SqlCommand("DELETE  FROM  Register
WHERE        (Employee_Code = '" + txtEmpCode.Text + "')", con);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("your data is Deleted");
            this.Refresh();
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        }
    }
```

Human Resource Management with C#: Edit Job Details (Admin)



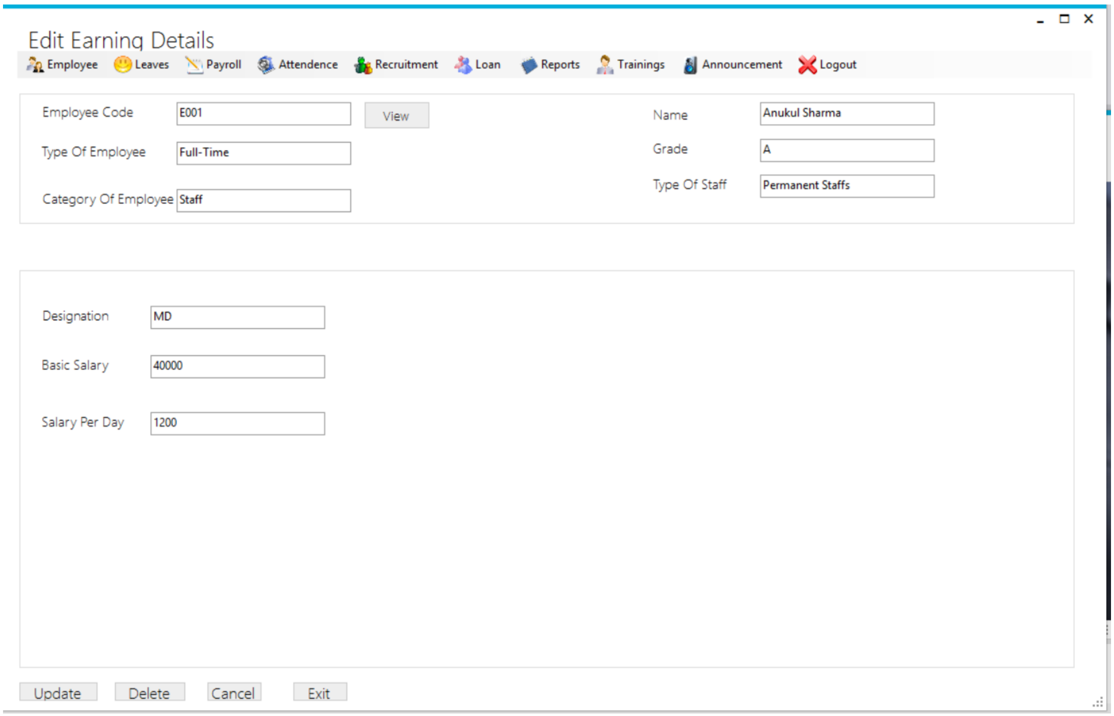
Employee Code: E001
Type Of Employee: Full-Time
Category Of Employee: Staff
Name: Anukul Sharma
Grade: A
Type Of Staff: Permanent Staffs

Mode Of Recruitment: Promotion
Date Of Joining: Saturday, December 5, 2015
Date Of Conformation: Thursday, December 10, 201
Date Of Last Increment / Promotio: Sunday, December 20, 2015
Branch: IT
Department: Management
Designation: MD
Reporting Boss: None
C.V. Uploaded: Yes

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;
```

```
namespace Human_Resourse_Management
{
    public partial class Edit_Job_Details : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Edit_Job_Details()
        {
            InitializeComponent();
        }
         public Edit_Job_Details(int ID)
        {
            InitializeComponent();
            btnUpdate.Text = "Update";
            btnDelete.Text = "Delete";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Job_Details where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }

        }
        private void Edit_Job_Details_Load(object sender, EventArgs
e)
        {

        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            foreach (Control c in groupBox1.Controls)
            {
                txtEmpCode.Text = "";
                txtName.Text = "";
                comboGrade.Text = "";
                comboCategoryOfEmployee.Text = "";
                comboTypeOfEmployee.Text = "";
                comboTypeOfStaff.Text = "";

            }
            foreach (Control b in groupBox2.Controls)
            {
                comboModeOfRecruitment.Text = "";
                comboReportingBoss.Text = "";
                txtBranch.Text = "";
                comboCVUploaded.Text = "";
                txtDesignation.Text = "";
                comboDepartment.Text = "";
                DateTime_DateOfJoining.Text = "";
                DateTime_DateOfConformation.Text = "";
                DateTime_DateOfLastIncement.Text = "";

            }
        }
        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void logoutToolStripMenuItem_Click(object sender, 
EventArgs e)
        {
            this.Close();
        }

        private void btnUpdate_Click(object sender, EventArgs e)
        {
            if (txtEmpCode.Text != "")
            {
                SqlConnection con = new 
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("UPDATE Job_Details 
SET Name='" + this.txtName.Text + "',Type_Of_Employee='" + 
this.comboTypeOfEmployee.Text + "',Grade='" + this.comboGrade.Text + 
"',Category_Of_Employee='" + this.comboCategoryOfEmployee.Text + 
"',Type_Of_Staff='" + this.comboTypeOfStaff.Text + 
"',Mode_Of_Recruitment='" + this.comboModeOfRecruitment.Text + 
"',Date_Of_Joining='" + this.DateTime_DateOfJoining.Text + 
"',Date_Of_Conformation='" + this.DateTime_DateOfConformation.Text + 
"',Date_Of_Last_Increment='" + this.DateTime_DateOfLastIncement.Text 
+ "',Branch='" + this.txtBranch.Text + "',Department='" + 
this.comboDepartment.Text + "',Designation='" + 
this.txtDesignation.Text + "',Reporting_Boss='" + 
this.comboReportingBoss.Text + "',CV_Uploaded='" + 
this.comboCVUploaded.Text + "' WHERE Employee_Code='" + 
this.txtEmpCode.Text + "'", con);
                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                }
                catch (SqlException ex) { throw ex; }
                finally { con.Close(); }
                MessageBox.Show("Data is Updated");
                this.Refresh();


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }



        }

        private void btnDelete_Click(object sender, EventArgs e)
        {
            SqlConnection con = new 
SqlConnection(db.connectionString);
            SqlCommand cmd = new SqlCommand("DELETE  FROM 
Job_Details  WHERE        (Employee_Code = '" + txtEmpCode.Text + 
"')", con);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("your data is Deleted");
            this.Refresh();
        }

        private void btnView_Click(object sender, EventArgs e)
        {
            if (txtEmpCode.Text != "")
            {
                try
                {
                    SqlConnection con = new
SqlConnection(db.connectionString);
                    string sql = "SELECT *  FROM   Job_Details WHERE
Employee_Code='" + this.txtEmpCode.Text + "'";
                    SqlCommand cmd1 = new SqlCommand(sql, con);
                    SqlDataAdapter adp = new SqlDataAdapter();
                    adp.SelectCommand = cmd1;
                    DataTable dt = new DataTable();
                    adp.Fill(dt);
                    // now get the values
                    txtName.Text = dt.Rows[0][2].ToString();
                    comboTypeOfEmployee.Text =
dt.Rows[0][3].ToString();
                    comboGrade.Text = dt.Rows[0][4].ToString();
                    comboCategoryOfEmployee.Text =
dt.Rows[0][5].ToString();
                    comboTypeOfStaff.Text = dt.Rows[0][6].ToString();
                    comboModeOfRecruitment.Text =
dt.Rows[0][7].ToString();
                    DateTime_DateOfJoining.Text =
dt.Rows[0][8].ToString();
                    DateTime_DateOfConformation.Text =
dt.Rows[0][9].ToString();
                    DateTime_DateOfLastIncement.Text =
dt.Rows[0][10].ToString();
                    txtBranch.Text = dt.Rows[0][11].ToString();
                    comboDepartment.Text = dt.Rows[0][12].ToString();
                    txtDesignation.Text = dt.Rows[0][13].ToString();
                    comboReportingBoss.Text =
dt.Rows[0][14].ToString();
                    comboCVUploaded.Text = dt.Rows[0][15].ToString();

                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.ToString());
                }
            }
        }
    }
}
```

Human Resource Management with C#: Edit Earning Details (Admin)



Employee Code: E001

Type Of Employee: Full-Time

Category Of Employee: Staff

Name: Anukul Sharma

Grade: A

Type Of Staff: Permanent Staffs

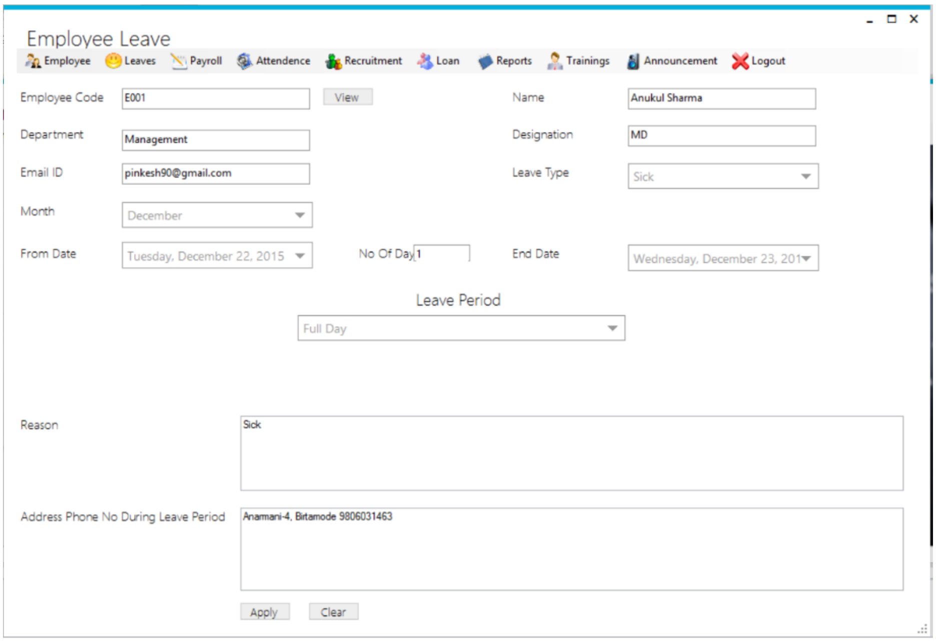
Designation: MD

Basic Salary: 40000

Salary Per Day: 1200

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    public partial class Edit_Earning_Details : MetroForm
    {
        Human_Resource_Management.Database db = new
Human_Resource_Management.Database();
        public Edit_Earning_Details()
        {
            InitializeComponent();
        }
        public Edit_Earning_Details(int ID)
        {
            InitializeComponent();
            btnUpdate.Text = "Update";
            btnDelete.Text = "Delete";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Earning_Details where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }

        }
        private void Edit_Earning_Details_Load(object sender,
EventArgs e)
        {

        }

        private void editEmployeeToolStripMenuItem2_Click(object
sender, EventArgs e)
        {

        }

        private void addNewEmployeeToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Register().Show();
            this.Hide();
        }

        private void editEmployeeToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Edit_Personal_Details().Show();
            this.Hide();
        }

        private void editEmployeeToolStripMenuItem1_Click(object
sender, EventArgs e)
        {
            new Edit_Job_Details().Show();
            this.Hide();
        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            foreach (Control c in groupBox1.Controls)
            {
                txtEmpCode.Text = "";
                txtName.Text = "";
                comboCategoryOfEmployee.Text = "";
                ComboGrade.Text = "";
                comboTypeOfEmployee.Text = "";
                comboTypeOfStaff.Text = "";
            }
            foreach (Control b in groupBox2.Controls)
            {
                txtDesignation.Text = "";
                txtBasicSalary.Text = "";
                txtSalaryPerDay.Text = "";
            }
        }

        private void btnUpdate_Click(object sender, EventArgs e)
        {
            if (txtName.Text != "")
            {
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("UPDATE
Earning_Details SET Name='" + this.txtName.Text +
"',Type_Of_Employee='" + this.comboTypeOfEmployee.Text + "',Grade='"
+ this.ComboGrade.Text + "',Category_Of_Employee='" +
this.comboCategoryOfEmployee.Text + "',Type_Of_Staff='" +
this.comboTypeOfStaff.Text + "',Designation='" +
this.txtDesignation.Text + "',Basic_Salary='" +
this.txtBasicSalary.Text + "',Salary_Per_Day='" +
this.txtSalaryPerDay.Text + "' WHERE  Employee_Code='" +
this.txtEmpCode.Text + "'", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Updated");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void btnDelete_Click(object sender, EventArgs e)
        {
            SqlConnection con = new
SqlConnection(db.connectionString);
            SqlCommand cmd = new SqlCommand("DELETE  FROM
Earning_Details  WHERE        (Employee_Code = '" + txtEmpCode.Text +
"')", con);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("your data is Deleted");
            this.Refresh();
        }

        private void btnView_Click(object sender, EventArgs e)
        {
            if (txtEmpCode.Text != "")
            {
                try
                {
                    SqlConnection con = new
SqlConnection(db.connectionString);
                    string sql = "SELECT *  FROM   Earning_Details
WHERE Employee_Code='" + this.txtEmpCode.Text + "'";
                    SqlCommand cmd1 = new SqlCommand(sql, con);
                    SqlDataAdapter adp = new SqlDataAdapter();
                    adp.SelectCommand = cmd1;
                    DataTable dt = new DataTable();
                    adp.Fill(dt);
                    // now get the values
                    txtDesignation.Text = dt.Rows[0][2].ToString();
                    txtName.Text = dt.Rows[0][3].ToString();
                    comboTypeOfEmployee.Text =
dt.Rows[0][4].ToString();
                    comboCategoryOfEmployee.Text =
dt.Rows[0][5].ToString();
                    ComboGrade.Text = dt.Rows[0][6].ToString();
                    comboTypeOfStaff.Text = dt.Rows[0][7].ToString();
                    txtBasicSalary.Text = dt.Rows[0][8].ToString();
                    txtSalaryPerDay.Text = dt.Rows[0][9].ToString();
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.ToString());
                }
            }
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void txtEmpCode_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }

        private void txtName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }
    }
}
```

Human Resource Management with C#: Employee Leave (Admin)



Employee Code: E001

Name: Anukul Sharma

Department: Management

Designation: MD

Email ID: pinkesh90@gmail.com

Leave Type: Sick

Month: December

From Date: Tuesday, December 22, 2015

No Of Day: 1

End Date: Wednesday, December 23, 201

Leave Period: Full Day

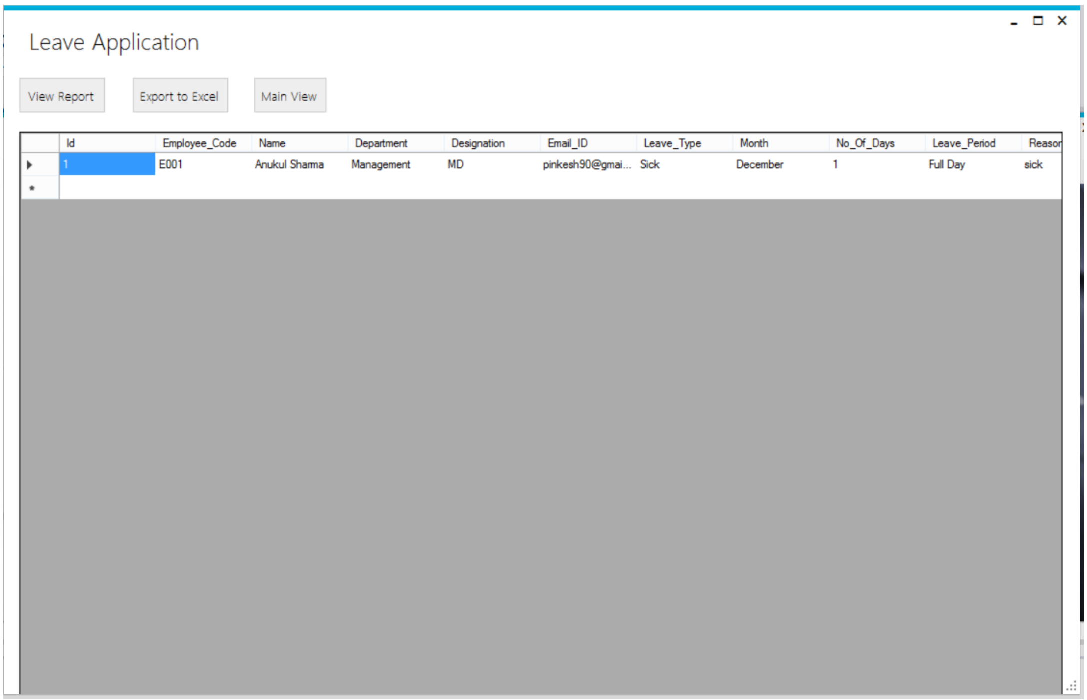
Reason: Sick

Address Phone No During Leave Period: Anamani-4, Birtamode 9806031463

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management.Admin
{
    public partial class Employee_Leave : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Employee_Leave()
        {
            InitializeComponent();
        }
        public Employee_Leave(int ID)
        {
            InitializeComponent();
            btnApply.Text = "Apply";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Employee_Leave where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }
        }
        private void Employee_Leave_Load(object sender, EventArgs e)
        {

        }

        private void btnApply_Click(object sender, EventArgs e)
        {
            if (txtName.Text != "")
            {
                string ds1 = DateTime_FromDate.Text;
                string ds2 = DateTime_EndDate.Text;
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Employee_Leave
(Employee_Code,Name,Department,Designation,Email_ID,Leave_Type,Month,
No_Of_Days,Leave_Period,Reason,Address,From_Date,End_Date) VALUES('"
+ txtEmpCode.Text + "','" + txtName.Text + "','" +
comboDepartment.Text + "','" + txtDesignation.Text + "','" +
txtEmail.Text + "','" + comboLeaveType.Text + "','" + comboMonth.Text
+ "','" + txtNoDays.Text + "','" + comboLeavePeriod.Text + "','" +
txtReason.Text + "','" + txtAddress.Text + "','" +
DateTime_FromDate.Text + "','" + DateTime_EndDate.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Sent");
            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void txtEmpCode_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }
        private void txtEmail_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsEmailId(e);
        }

        private void txtNoDays_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsQuantity(e);
        }

        private void btnView_Click(object sender, EventArgs e)
        {

            try
            {
                SqlConnection con = new
SqlConnection(db.connectionString);
                string sql = "SELECT *  FROM   Employee_Leave WHERE
Employee_Code='" + this.txtEmpCode.Text + "'";
                SqlCommand cmd1 = new SqlCommand(sql, con);
                SqlDataAdapter adp = new SqlDataAdapter();
                adp.SelectCommand = cmd1;
                DataTable dt = new DataTable();
                adp.Fill(dt);
                // now get the values
                txtName.Text = dt.Rows[0][2].ToString();
                comboDepartment.Text = dt.Rows[0][3].ToString();
                txtDesignation.Text = dt.Rows[0][4].ToString();
            }
             catch (Exception ex)
                {
                    MessageBox.Show(ex.ToString());
                }
            }


        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }
    }
```

```
}
```

Human Resource Management with C#: Leave Application (Admin)



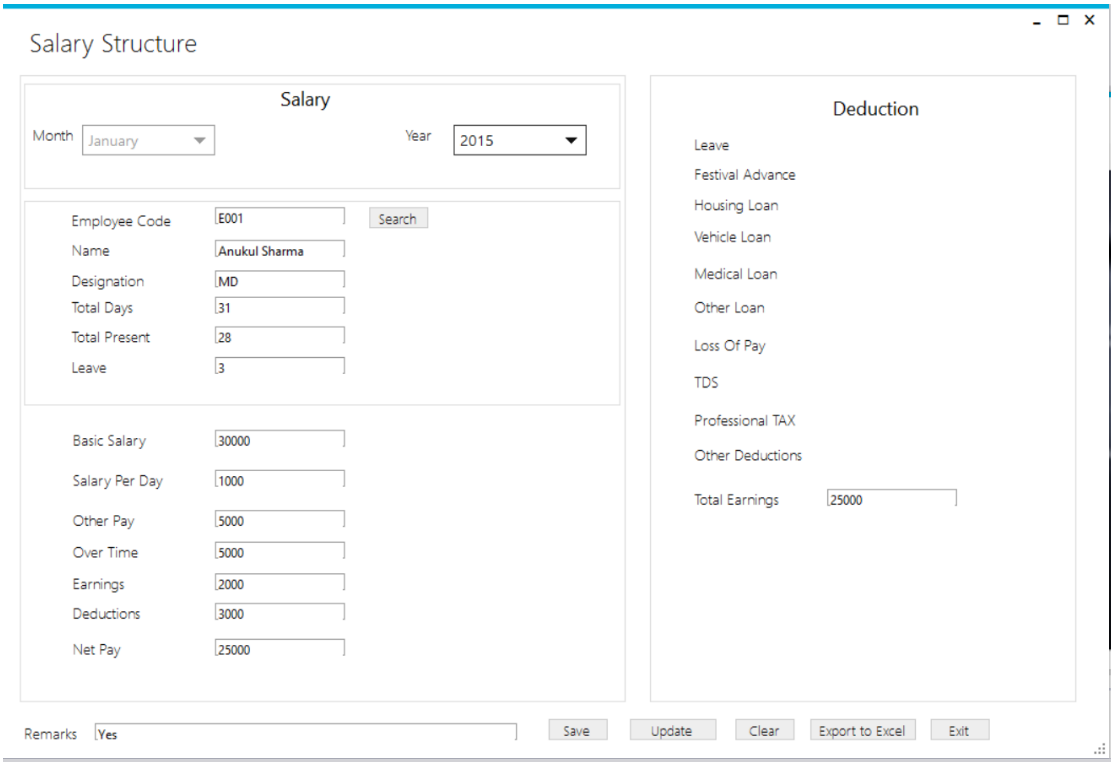
	Id	Employee_Code	Name	Department	Designation	Email_ID	Leave_Type	Month	No_Of_Days	Leave_Period	Reason
▸	1	E001	Anukul Sharma	Management	MD	pinkesh90@gmai...	Sick	December	1	Full Day	sick
*											

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    public partial class Leave_Application : MetroForm
    {
        public Leave_Application()
        {
            InitializeComponent();
        }

        private void Leave_Application_Load(object sender, EventArgs
e)
        {
            // TODO: This line of code loads data into the
'cHRMSDataSet.Employee_Leave' table. You can move, or remove it, as
needed.
```

```
this.employee_LeaveTableAdapter.Fill(this.cHRMSDataSet.Employee_Leave
);

        }

        private void viewDetails_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
        {

        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }

        private void btnView_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = @"Data Source=Anukul-
pc\sqlexpress;Initial Catalog=CHRMS;Integrated Security=True";
            SqlCommand command = new SqlCommand();
            command.Connection = con;
            command.CommandText = "SELECT * FROM  Employee_Leave";
            DataTable data = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            adapter.Fill(data);
            DataGridView.DataSource = data;
        }


        private void btnExport_Click(object sender, EventArgs e)
        {
            // Creating a Excel object.
            Microsoft.Office.Interop.Excel._Application excel = new
Microsoft.Office.Interop.Excel.Application();
            Microsoft.Office.Interop.Excel._Workbook workbook =
excel.Workbooks.Add(Type.Missing);
            Microsoft.Office.Interop.Excel._Worksheet worksheet =
null;

            try
            {

                worksheet = workbook.ActiveSheet;

                worksheet.Name = "ExportedFromDatGrid";

                int cellRowIndex = 1;
                int cellColumnIndex = 1;

                //Loop through each row and read value from each
column.
                for (int i = 0; i < DataGridView.Rows.Count - 1; i++)
                {
                    for (int j = 0; j < DataGridView.Columns.Count;
j++)
                    {
                        // Excel index starts from 1,1. As first Row
would have the Column headers, adding a condition check.
```

```
                        if (cellRowIndex == 1)
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = DataGridView.Columns[j].HeaderText;
                        }
                        else
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = DataGridView.Rows[i].Cells[j].Value.ToString();
                        }
                        cellColumnIndex++;
                    }
                    cellColumnIndex = 1;
                    cellRowIndex++;
                }

                //Getting the location and file name of the excel to
save from user.
                SaveFileDialog saveDialog = new SaveFileDialog();
                saveDialog.Filter = "Excel files (*.xlsx)|*.xlsx|All
files (*.*)|*.*";
                saveDialog.FilterIndex = 2;

                if (saveDialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
                {
                    workbook.SaveAs(saveDialog.FileName);
                    MessageBox.Show("Export Successful");
                }
            }
            catch (System.Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                excel.Quit();
                workbook = null;
                excel = null;
            }
        }

        private void btnMainView_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Human Resource Management with C#: Salary Structure (Admin)



Salary Structure

Salary

Month: January

Year: 2015

Employee Code	E001
Name	Anukul Sharma
Designation	MD
Total Days	31
Total Present	28
Leave	3
Basic Salary	30000
Salary Per Day	1000
Other Pay	5000
Over Time	5000
Earnings	2000
Deductions	3000
Net Pay	25000

Deduction

Leave
Festival Advance
Housing Loan
Vehicle Loan
Medical Loan
Other Loan
Loss Of Pay
TDS
Professional TAX
Other Deductions

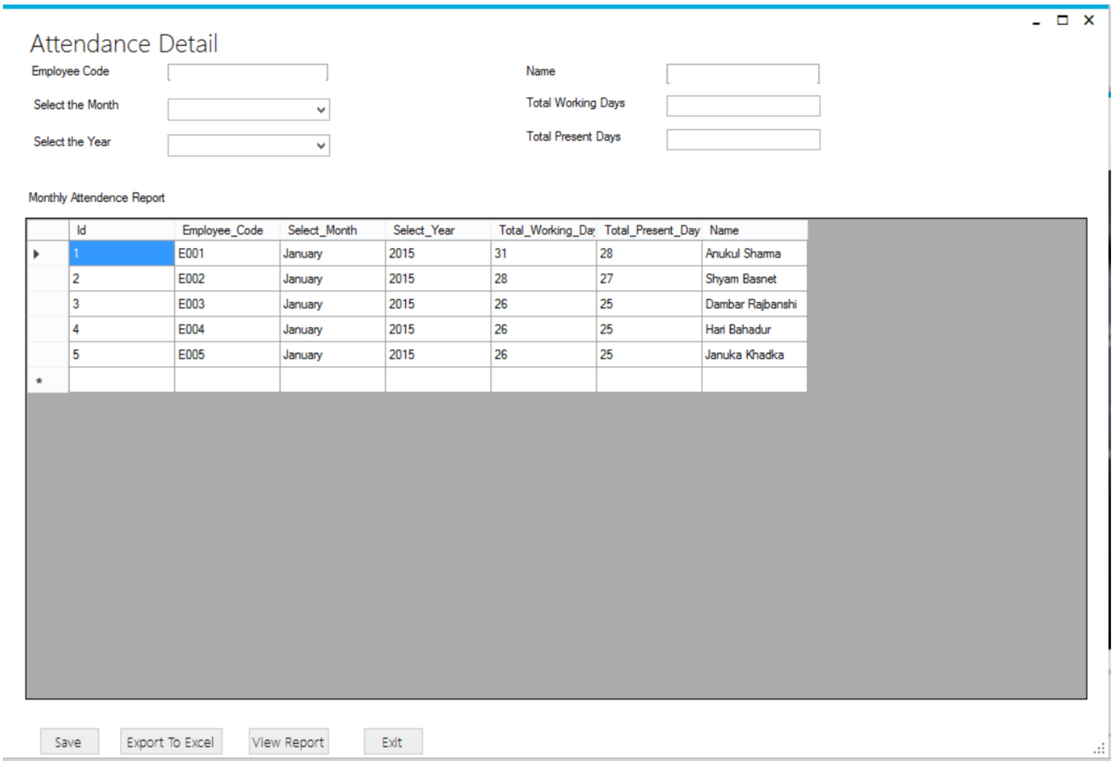
Total Earnings: 25000

Remarks: Yes

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;
using ExcelLibrary.CompoundDocumentFormat;
using ExcelLibrary.SpreadSheet;
using ExcelLibrary.BinaryFileFormat;
using ExcelLibrary.BinaryDrawingFormat;
using Microsoft.Office.Interop.Excel;
using Excel = Microsoft.Office.Interop.Excel;

namespace Human_Resourse_Management
{
    public partial class Salary_Structure : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Salary_Structure()
        {
            InitializeComponent();
        }
        public Salary_Structure(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            btnUpdate.Text = "Update";
            btnClear.Text = "CLear";
            btnExit.Text = "Exit";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Salary_Structure where Name ='" + ID + "'", con);
                System.Data.DataTable dt = new
System.Data.DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }
        }
        private void Salary_Structure_Load(object sender, EventArgs
e)
        {

        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtEmpCode.Text != "")
            {
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Salary_Structure
(Employee_Code,Name,Designation,Total_Days,Total_Present,Leave,Basic_
Salary,Salary_Per_Day,Other_Pay,Over_Time,Earnings,Deduction,Net_Pay,
Month,Year,Remarks,Total_Earnings) VALUES('" + txtEmpCode.Text +
"','" + txtName.Text + "','" + txtDesignation.Text + "','" +
txtTotalDays.Text + "','" + txtTotalPresent.Text + "','" +
txtLeave.Text + "','" + txtBasicSalary.Text + "','" +
txtSalaryPerDay.Text + "','" + txtOtherPay.Text + "','" +
txtOverTime.Text + "','" + txtEarnings.Text + "','" +
txtDeduction.Text + "','" + txtNetPay.Text + "','" + comboMonth.Text
+ "','" + comboYear.Text + "','" + txtRemarks.Text + "','" +
txtTotalEarnings.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Saved");
            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void btnClear_Click(object sender, EventArgs e)
        {
            foreach (Control c in groupBox1.Controls)
            {
                c.Text = "";

            }
            foreach (Control b in groupBox2.Controls)
            {
                b.Text = "";
            }
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnUpdate_Click(object sender, EventArgs e)
        {
            if (txtEmpCode.Text != "")
            {
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("UPDATE
Salary_Structure SET Employee_Code='" + txtEmpCode.Text + "', Name='"
+ txtName.Text + "',Designation='" + txtDesignation.Text +
"',Total_Days='" + txtTotalDays.Text + "',Total_Present='" +
txtTotalPresent.Text + "',Leave='" + txtLeave.Text +
"',Basic_Salary='" + txtBasicSalary.Text + "',Salary_Per_Day='" +
txtSalaryPerDay.Text + "',Other_Pay='" + txtOtherPay.Text +
"',Over_Time='" + txtOverTime.Text + "',Earniings='" +
txtEarnings.Text + "',Deduction='" + txtDeduction.Text +
"',Net_Pay='" + txtNetPay.Text + "',Month='" + comboMonth.Text +
"',Year='" + comboYear.Text + "',Remarks='" + txtRemarks.Text +
"',Total_Earnings='" + txtTotalEarnings.Text + "'),", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Updated");

            }
        }

        private void btnExportToExcel_Click(object sender, EventArgs
e)
        {

            SqlConnection cnn;
            string connectionstring = null;
            string sql = null;
            string data = null;
            int i = 0;
            int j = 0;

            Excel.Application xlApp;
            Excel.Workbook xlWorkBook;
            Excel.Worksheet xlWorkSheet;
            object misValue = System.Reflection.Missing.Value;


            xlApp = new Microsoft.Office.Interop.Excel.Application();
            xlWorkBook = xlApp.Workbooks.Add(misValue);
            xlWorkSheet =
(Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);

            connectionstring = @"Data Source=Anukul-
pc\sqlexpress;Initial Catalog=CHRMS;Integrated Security=True";
            cnn = new SqlConnection(connectionstring);
            cnn.Open();
            sql = "SELECT * FROM Salary_Structure";
            SqlDataAdapter dscmd = new SqlDataAdapter(sql, cnn);
            DataSet ds = new DataSet();
            dscmd.Fill(ds);

            for (i = 0; i <= ds.Tables[0].Rows.Count - 1; i++)
            {
                for (j = 0; j <= ds.Tables[0].Columns.Count - 1; j++)
                {
                    data =
ds.Tables[0].Rows[i].ItemArray[j].ToString();
                    xlWorkSheet.Cells[i + 1, j + 1] = data;
                }
            }


            xlWorkBook.SaveAs("informations.xls",
Excel.XlFileFormat.xlWorkbookNormal, misValue, misValue, misValue,
misValue, Excel.XlSaveAsAccessMode.xlExclusive, misValue, misValue,
misValue, misValue, misValue);
            xlWorkBook.Close(true, misValue, misValue);
            xlApp.Quit();

            releaseObject(xlWorkSheet);
            releaseObject(xlWorkBook);
            releaseObject(xlApp);


            MessageBox.Show("Excel file created , you can find the
file D:\\Sam-informations.xls");
        }

        private void releaseObject(object obj)
        {
            try
            {

System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
                obj = null;
            }
            catch (Exception ex)
            {
                obj = null;
                MessageBox.Show("Exception Occured while releasing
object " + ex.ToString());
            }
            finally
            {
                GC.Collect();
            }

        }

        private void btnSearch_Click(object sender, EventArgs e)
        {
            if (txtEmpCode.Text != "")
            {
                try
                {
                    SqlConnection con = new
SqlConnection(db.connectionString);
```

```
                    string sql = "SELECT *  FROM   Salary_Structure
WHERE Employee_Code='" + this.txtEmpCode.Text + "'";
                    SqlCommand cmd1 = new SqlCommand(sql, con);
                    SqlDataAdapter adp = new SqlDataAdapter();
                    adp.SelectCommand = cmd1;
                    System.Data.DataTable dt = new
System.Data.DataTable();
                    adp.Fill(dt);
                    // now get the values
                    txtName.Text = dt.Rows[0][2].ToString();
                    txtDesignation.Text = dt.Rows[0][3].ToString();
                    txtTotalDays.Text = dt.Rows[0][4].ToString();
                    txtTotalPresent.Text = dt.Rows[0][5].ToString();
                    txtLeave.Text = dt.Rows[0][6].ToString();
                    txtBasicSalary.Text = dt.Rows[0][7].ToString();
                    txtSalaryPerDay.Text = dt.Rows[0][8].ToString();
                    txtOtherPay.Text = dt.Rows[0][9].ToString();
                    txtOverTime.Text = dt.Rows[0][10].ToString();
                    txtEarnings.Text = dt.Rows[0][11].ToString();
                    txtDeduction.Text = dt.Rows[0][12].ToString();
                    txtNetPay.Text = dt.Rows[0][13].ToString();
                    comboMonth.Text = dt.Rows[0][14].ToString();
                    comboYear.Text = dt.Rows[0][15].ToString();
                    txtRemarks.Text = dt.Rows[0][16].ToString();
                    txtTotalEarnings.Text =
dt.Rows[0][17].ToString();



                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.ToString());
                }
            }

        }



        private void txtBasicSalary_KeyPress(object sender,
KeyPressEventArgs e)
        {


        }
    }
}
```

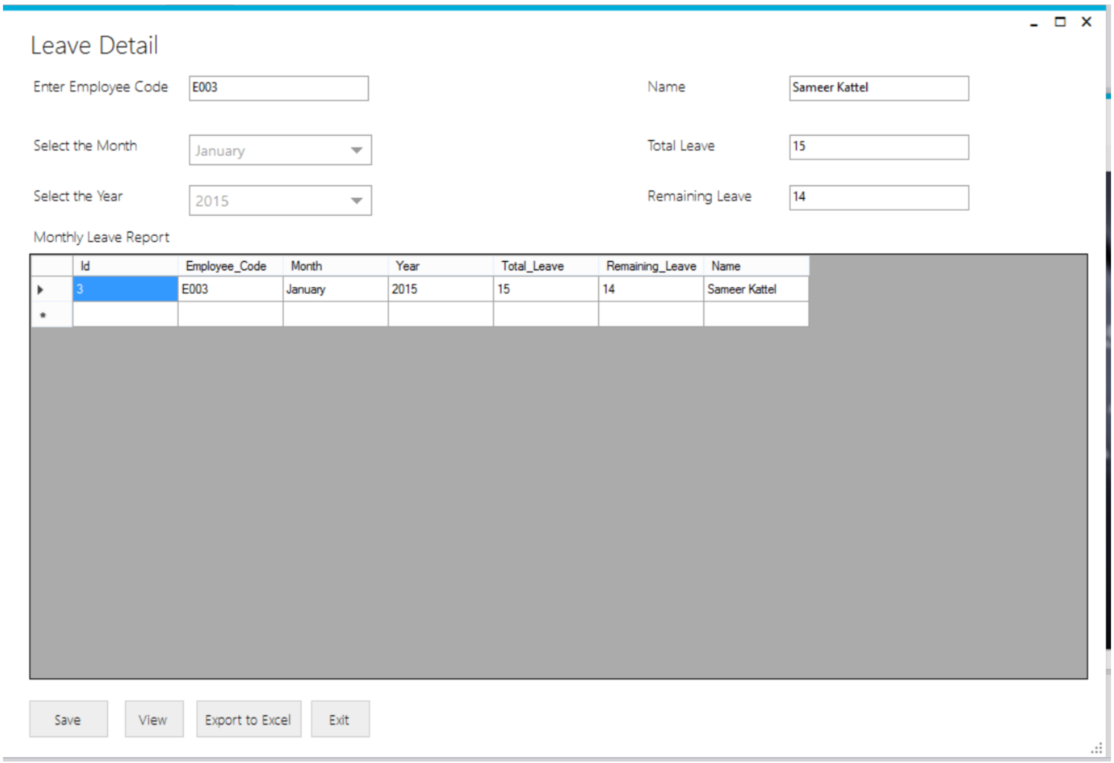
Human Resource Management with C#: Attendance Detail (Admin)



	Id	Employee_Code	Select_Month	Select_Year	Total_Working_Da	Total_Present_Day	Name
▸	1	E001	January	2015	31	28	Anukul Sharma
	2	E002	January	2015	28	27	Shyam Basnet
	3	E003	January	2015	26	25	Dambar Rajbanshi
	4	E004	January	2015	26	25	Hari Bahadur
	5	E005	January	2015	26	25	Januka Khadka
*							

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;
using ExcelLibrary.CompoundDocumentFormat;
using ExcelLibrary.SpreadSheet;
using ExcelLibrary.BinaryFileFormat;
using ExcelLibrary.BinaryDrawingFormat;

namespace Human_Resourse_Management.Admin
{
    public partial class Employees_Monthly_Attendance : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Employees_Monthly_Attendance()
        {
            InitializeComponent();
        }
        public Employees_Monthly_Attendance(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Employee_Monthly_Attendence where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmployeeCode.Text = dt.Rows[0][0].ToString();
                    txtName1.Text = dt.Rows[0][1].ToString();
                }
            }

        }

        private void Employees_Monthly_Attendance_Load(object sender,
EventArgs e)
        {

        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtName.Text != "")
            {
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Employee_Monthly_Attendence(Employee_Code,Name,Select_Month,Select_Ye
ar,Total_Working_Days,Total_Present_Days) VALUES('" +
txtEmployeeCode.Text + "','" + txtName1.Text + "','" +
comboSelectMonth.Text + "','" + comboSelectYear.Text + "','" +
txtTotalWorkingDays.Text + "','" + txtTotalPresentDays.Text + "')",
con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Saved");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void btnClear_Click(object sender, EventArgs e)
        {
            foreach (Control c in Controls)
            {
                txtEmployeeCode.Text = "";
                txtName1 .Text= "";
                comboSelectMonth.Text = "";
                comboSelectYear.Text = "";
                txtTotalPresentDays.Text = "";
                txtTotalWorkingDays.Text = "";
            }
        }


        private void btnViewReport_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = @"Data Source=Anukul-
pc\sqlexpress;Initial Catalog=CHRMS;Integrated Security=True";
            SqlCommand command = new SqlCommand();
            command.Connection = con;
            command.CommandText = "SELECT * FROM
Employee_Monthly_Attendence";
            DataTable data = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            adapter.Fill(data);
            dataGridView1.DataSource = data;
        }

        private void btnExportToExcel_Click(object sender, EventArgs
e)
        {

            // Creating a Excel object.
            Microsoft.Office.Interop.Excel._Application excel = new
Microsoft.Office.Interop.Excel.Application();
            Microsoft.Office.Interop.Excel._Workbook workbook =
excel.Workbooks.Add(Type.Missing);
            Microsoft.Office.Interop.Excel._Worksheet worksheet =
null;

            try
            {

                worksheet = workbook.ActiveSheet;

                worksheet.Name = "ExportedFromDatGrid";

                int cellRowIndex = 1;
                int cellColumnIndex = 1;

                //Loop through each row and read value from each
column.
                for (int i = 0; i < dataGridView1.Rows.Count - 1;
i++)
                {
                    for (int j = 0; j < dataGridView1.Columns.Count;
j++)
                    {
                        // Excel index starts from 1,1. As first Row
would have the Column headers, adding a condition check.
                        if (cellRowIndex == 1)
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = dataGridView1.Columns[j].HeaderText;
                        }
                        else
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = dataGridView1.Rows[i].Cells[j].Value.ToString();
                        }
                        cellColumnIndex++;
                    }
                    cellColumnIndex = 1;
                    cellRowIndex++;
                }

                //Getting the location and file name of the excel to
save from user.
                SaveFileDialog saveDialog = new SaveFileDialog();
                saveDialog.Filter = "Excel files (*.xlsx)|*.xlsx|All
files (*.*)|*.*";
                saveDialog.FilterIndex = 2;

                if (saveDialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
                {
                    workbook.SaveAs(saveDialog.FileName);
                    MessageBox.Show("Export Successful");
                }
            }
            catch (System.Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                excel.Quit();
                workbook = null;
                excel = null;
            }

        }

    }
        }
```

Human Resource Management with C#: Leave Detail (Admin)



Leave Detail

Enter Employee Code: E003

Name: Sameer Kattel

Select the Month: January

Total Leave: 15

Select the Year: 2015

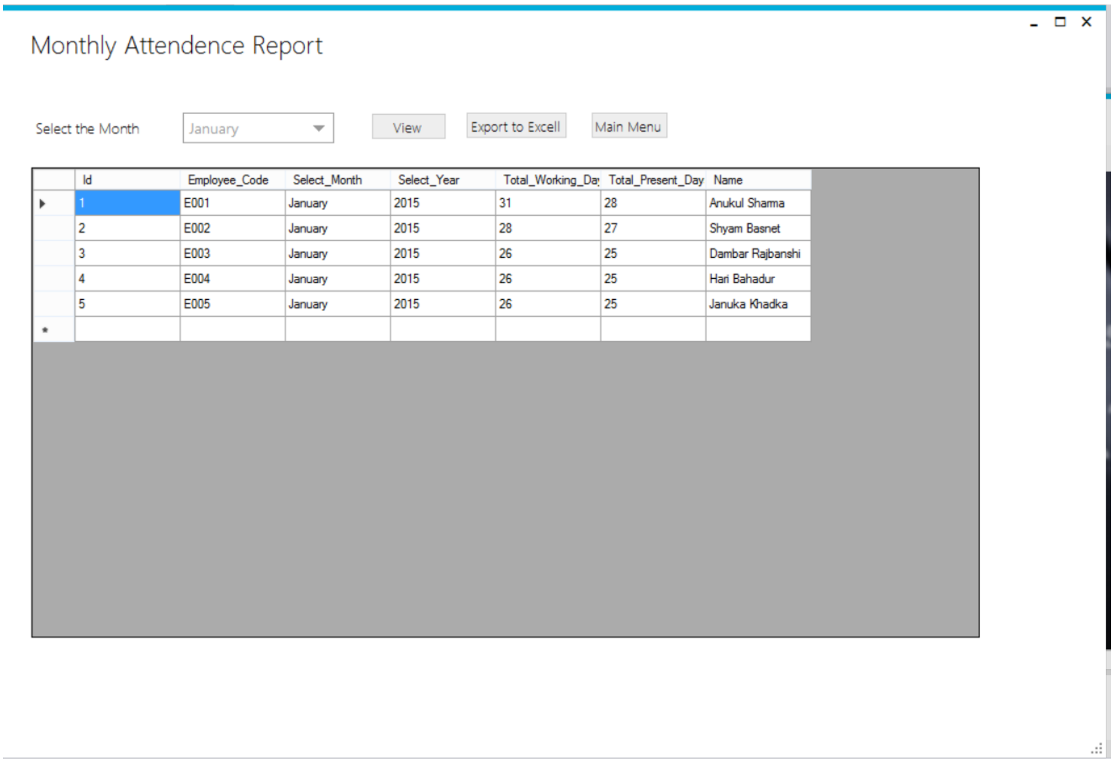
Remaining Leave: 14

Monthly Leave Report

	Id	Employee_Code	Month	Year	Total_Leave	Remaining_Leave	Name
▸	3	E003	January	2015	15	14	Sameer Kattel
*							

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;
using ExcelLibrary.CompoundDocumentFormat;
using ExcelLibrary.SpreadSheet;
using ExcelLibrary.BinaryFileFormat;
using ExcelLibrary.BinaryDrawingFormat;

namespace Human_Resourse_Management.Admin
{
    public partial class Monthly_Leave_Report : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Monthly_Leave_Report()
        {
            InitializeComponent();
        }
        public Monthly_Leave_Report(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Monthly_Leave_Report where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                }
            }

        }

        private void Monthly_Leave_Report_Load(object sender,
EventArgs e)
        {

        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
                if (txtEmpCode.Text!="")
            {

                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Monthly_Leave_Report(Employee_Code,Month,Year,Total_Leave,Remaining_L
eave,Name) VALUES('" + txtEmpCode.Text + "','" + comboMonth.Text +
"','" + comboYear.Text + "','" + txtTotalLeave.Text + "','" +
txtRemainingLeave.Text + "','" + txtName.Text + "')", con);
                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                }
                catch (SqlException ex) { throw ex; }
                finally { con.Close(); }
                MessageBox.Show("Congratulations!!!, Your Data is
Saved");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }



        }
        private void btnView_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = @"Data Source=Anukul-
pc\sqlexpress;Initial Catalog=CHRMS;Integrated Security=True";
            SqlCommand command = new SqlCommand();
            command.Connection = con;
            command.CommandText = "SELECT * FROM 
Monthly_Leave_Report";
            DataTable data = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            adapter.Fill(data);
            dataGridView1.DataSource = data;
        }

        private void btnExport_Click(object sender, EventArgs e)
        {
             // Creating a Excel object.
            Microsoft.Office.Interop.Excel._Application excel = new 
Microsoft.Office.Interop.Excel.Application();
            Microsoft.Office.Interop.Excel._Workbook workbook = 
excel.Workbooks.Add(Type.Missing);
            Microsoft.Office.Interop.Excel._Worksheet worksheet = 
null;

            try
            {

                worksheet = workbook.ActiveSheet;

                worksheet.Name = "ExportedFromDatGrid";

                int cellRowIndex = 1;
                int cellColumnIndex = 1;

                //Loop through each row and read value from each 
column.
                for (int i = 0; i < dataGridView1.Rows.Count - 1; 
i++)
                {
                    for (int j = 0; j < dataGridView1.Columns.Count; 
j++)
                    {
                        // Excel index starts from 1,1. As first Row 
would have the Column headers, adding a condition check.
                        if (cellRowIndex == 1)
                        {
                            worksheet.Cells[cellRowIndex, 
cellColumnIndex] = dataGridView1.Columns[j].HeaderText;
                        }
                        else
                        {
                            worksheet.Cells[cellRowIndex, 
cellColumnIndex] = dataGridView1.Rows[i].Cells[j].Value.ToString();
                        }
                        cellColumnIndex++;
                    }
                    cellColumnIndex = 1;
                    cellRowIndex++;
                }
                //Getting the location and file name of the excel to
save from user.
                SaveFileDialog saveDialog = new SaveFileDialog();
                saveDialog.Filter = "Excel files (*.xlsx)|*.xlsx|All
files (*.*)|*.*";
                saveDialog.FilterIndex = 2;

                if (saveDialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
                {
                    workbook.SaveAs(saveDialog.FileName);
                    MessageBox.Show("Export Successful");
                }
            }
            catch (System.Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                excel.Quit();
                workbook = null;
                excel = null;
            }


        }

        private void txtEmpCode_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }

        private void txtName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtRemainingLeave_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtTotalLeave_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }
    }
}
```

Human Resource Management with C#: Monthly Attendence Report (Admin)

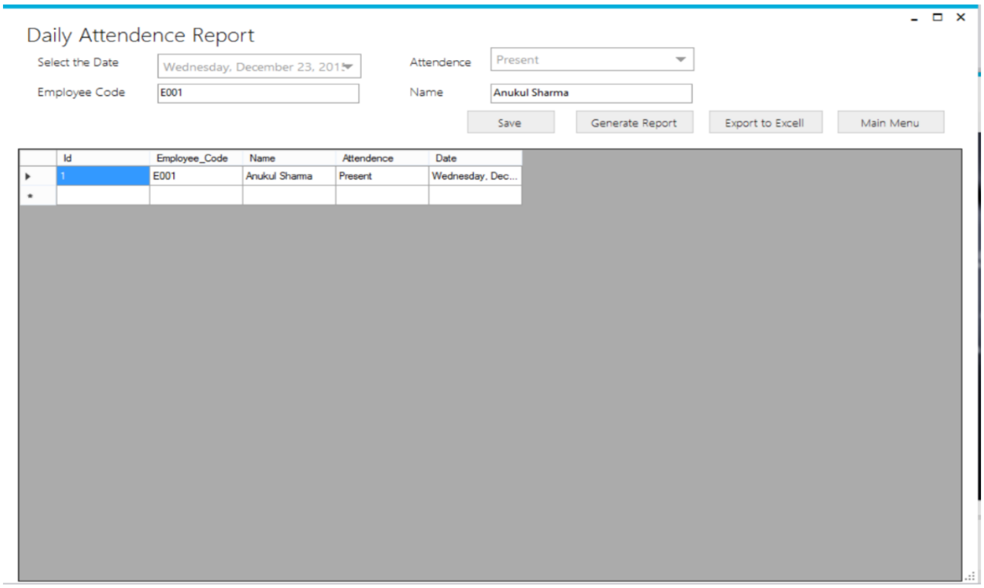


Id	Employee_Code	Select_Month	Select_Year	Total_Working_Day	Total_Present_Day	Name
1	E001	January	2015	31	28	Anukul Sharma
2	E002	January	2015	28	27	Shyam Basnet
3	E003	January	2015	26	25	Dambar Rajbanshi
4	E004	January	2015	26	25	Hari Bahadur
5	E005	January	2015	26	25	Januka Khadka

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;
using ExcelLibrary.SpreadSheet;
using ExcelLibrary.BinaryFileFormat;
using ExcelLibrary.BinaryDrawingFormat;


namespace Human_Resourse_Management.Admin
{
    public partial class Monthly_Attendence_Report : MetroForm
    {
         Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Monthly_Attendence_Report()
        {
            InitializeComponent();
        }

        private void Monthly_Attendence_Report_Load(object sender,
EventArgs e)
```

```
        {

        }

        private void metroButton4_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnView_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = @"Data Source=Anukul-
pc\sqlexpress;Initial Catalog=CHRMS;Integrated Security=True";
            SqlCommand command = new SqlCommand();
            command.Connection = con;
            command.CommandText = "SELECT * FROM
Employee_Monthly_Attendence";
            DataTable data = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            adapter.Fill(data);
            dataGridView1.DataSource = data;

        }

        private void ExportToExcel_Click(object sender, EventArgs e)
        {
            // Creating a Excel object.
            Microsoft.Office.Interop.Excel._Application excel = new
Microsoft.Office.Interop.Excel.Application();
            Microsoft.Office.Interop.Excel._Workbook workbook =
excel.Workbooks.Add(Type.Missing);
            Microsoft.Office.Interop.Excel._Worksheet worksheet =
null;

            try
            {

                worksheet = workbook.ActiveSheet;

                worksheet.Name = "ExportedFromDatGrid";

                int cellRowIndex = 1;
                int cellColumnIndex = 1;

                //Loop through each row and read value from each
column.
                for (int i = 0; i < dataGridView1.Rows.Count - 1;
i++)
                {
                    for (int j = 0; j < dataGridView1.Columns.Count;
j++)
                    {
                        // Excel index starts from 1,1. As first Row
would have the Column headers, adding a condition check.
                        if (cellRowIndex == 1)
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = dataGridView1.Columns[j].HeaderText;
                        }
                        else
                        {
                            worksheet.Cells[cellRowIndex, 
cellColumnIndex] = dataGridView1.Rows[i].Cells[j].Value.ToString();
                        }
                        cellColumnIndex++;
                    }
                    cellColumnIndex = 1;
                    cellRowIndex++;
                }

                //Getting the location and file name of the excel to 
save from user.
                SaveFileDialog saveDialog = new SaveFileDialog();
                saveDialog.Filter = "Excel files (*.xlsx)|*.xlsx|All 
files (*.*)|*.*";
                saveDialog.FilterIndex = 2;

                if (saveDialog.ShowDialog() == 
System.Windows.Forms.DialogResult.OK)
                {
                    workbook.SaveAs(saveDialog.FileName);
                    MessageBox.Show("Export Successful");
                }
            }
            catch (System.Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                excel.Quit();
                workbook = null;
                excel = null;
            }
        }
    }
  }
```

Human Resource Management with C#: Daily Attendence Report (Admin)



Daily Attendence Report

Select the Date: Wednesday, December 23, 2015

Attendence: Present

Employee Code: E001

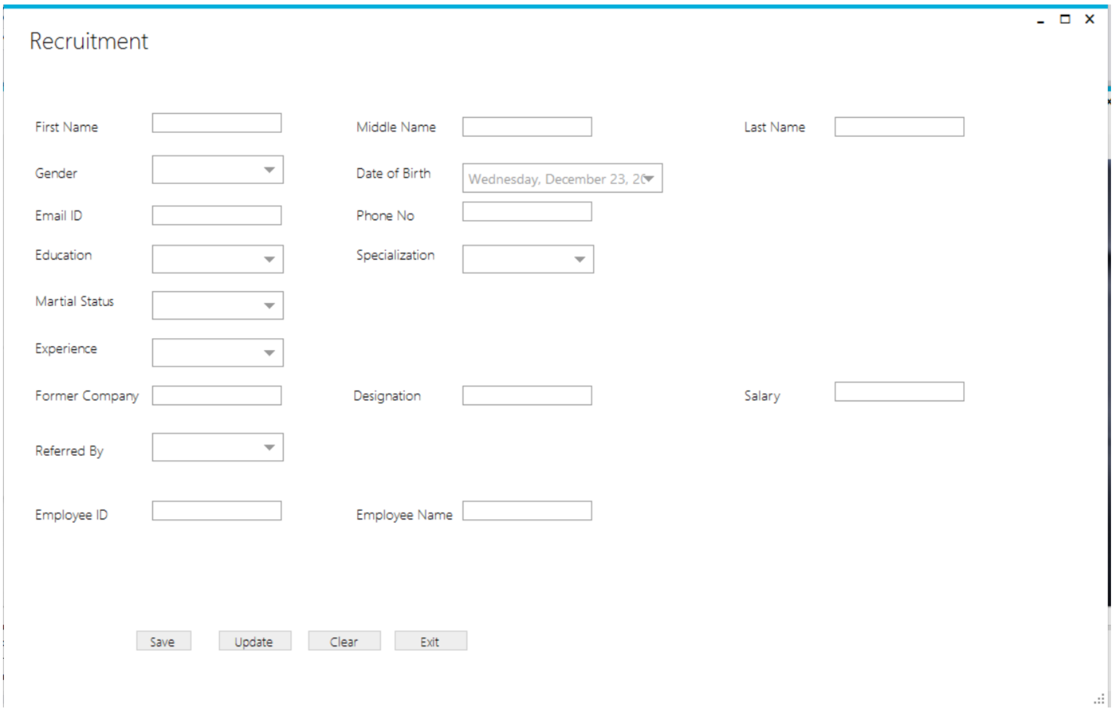
Name: Anukul Sharma

Id	Employee_Code	Name	Attendence	Date
1	E001	Anukul Sharma	Present	Wednesday, Dec...

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;
using ExcelLibrary.CompoundDocumentFormat;
using ExcelLibrary.SpreadSheet;
using ExcelLibrary.BinaryFileFormat;
using ExcelLibrary.BinaryDrawingFormat;

namespace Human_Resourse_Management.Admin
{
    public partial class Daily_Attendence_Report : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Daily_Attendence_Report()
        {
            InitializeComponent();
        }
        public Daily_Attendence_Report(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Daily_Attendence_Report  where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][1].ToString();
                    txtName.Text = dt.Rows[0][2].ToString();
                }
            }
        }
        private void Daily_Attendence_Report_Load(object sender,
EventArgs e)
        {

        }

        private void metroButton4_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtEmpCode.Text != "")
            {
                string dt = DateTime_Day.Text;
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Daily_Attendence_Report(Employee_Code,Name,Attendence,Date) VALUES('"
+ txtEmpCode.Text + "','" + txtName.Text + "','" +
comboAttendence.Text + "','" + DateTime_Day.Text + "')", con);
                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                }
                catch (SqlException ex) { throw ex; }
                finally { con.Close(); }
                MessageBox.Show("Congratulations!!!, Your Data is
Saved");
            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void btnGenerateReport_Click(object sender, EventArgs
e)
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = @"Data Source=Anukul-
pc\sqlexpress;Initial Catalog=CHRMS;Integrated Security=True";
            SqlCommand command = new SqlCommand();
            command.Connection = con;
            command.CommandText = "SELECT * FROM
Daily_Attendence_Report";
            DataTable data = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            adapter.Fill(data);
            dataGridView1.DataSource = data;
        }

        private void btnExport_Click(object sender, EventArgs e)
        {
             // Creating a Excel object.
            Microsoft.Office.Interop.Excel._Application excel = new
Microsoft.Office.Interop.Excel.Application();
            Microsoft.Office.Interop.Excel._Workbook workbook =
excel.Workbooks.Add(Type.Missing);
            Microsoft.Office.Interop.Excel._Worksheet worksheet =
null;

            try
            {

                worksheet = workbook.ActiveSheet;

                worksheet.Name = "ExportedFromDatGrid";

                int cellRowIndex = 1;
                int cellColumnIndex = 1;

                //Loop through each row and read value from each
column.
                for (int i = 0; i < dataGridView1.Rows.Count - 1;
i++)
```

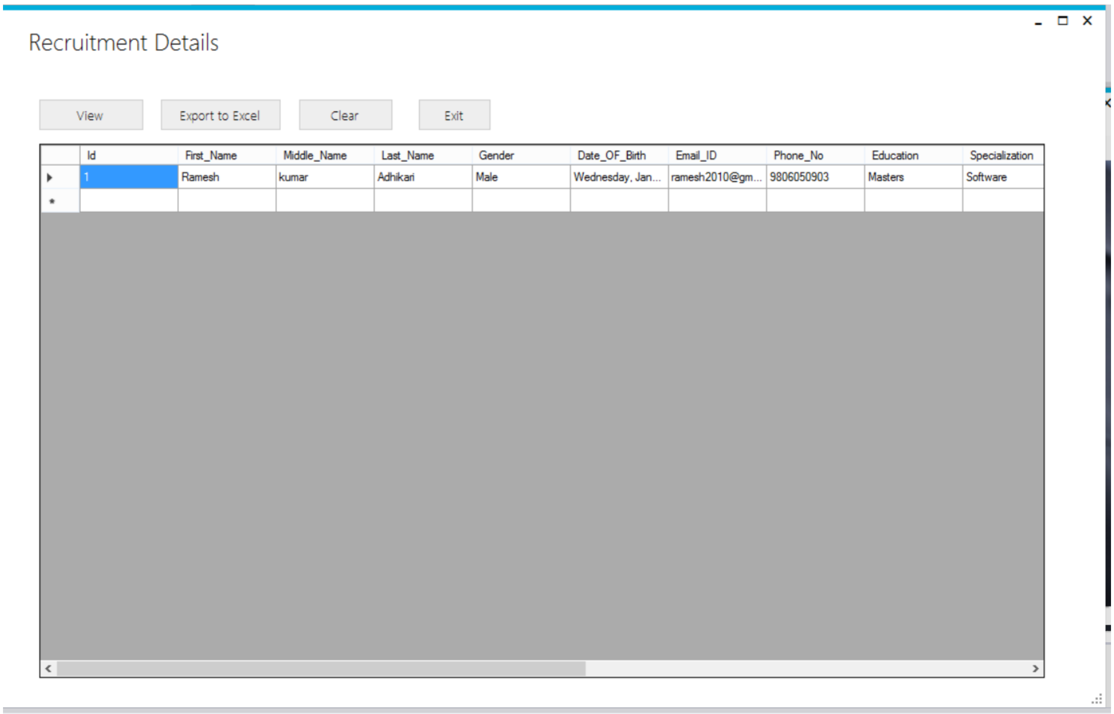
```
                {
                    for (int j = 0; j < dataGridView1.Columns.Count;
j++)
                    {
                        // Excel index starts from 1,1. As first Row
would have the Column headers, adding a condition check.
                        if (cellRowIndex == 1)
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = dataGridView1.Columns[j].HeaderText;
                        }
                        else
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = dataGridView1.Rows[i].Cells[j].Value.ToString();
                        }
                        cellColumnIndex++;
                    }
                    cellColumnIndex = 1;
                    cellRowIndex++;
                }

                //Getting the location and file name of the excel to
save from user.
                SaveFileDialog saveDialog = new SaveFileDialog();
                saveDialog.Filter = "Excel files (*.xlsx)|*.xlsx|All
files (*.*)|*.*";
                saveDialog.FilterIndex = 2;

                if (saveDialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
                {
                    workbook.SaveAs(saveDialog.FileName);
                    MessageBox.Show("Export Successful");
                }
            }
            catch (System.Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                excel.Quit();
                workbook = null;
                excel = null;
            }

        }
        }
    }
```

Human Resource Management with C#: Recruitment (Admin)



```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management.Admin
{
    public partial class Recruitment : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Recruitment()
        {
            InitializeComponent();
        }
        public Recruitment(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Register where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                  txtFirstName.Text = dt.Rows[0][0].ToString();
                    txtMiddleName.Text = dt.Rows[0][1].ToString();
                }
            }

        }

        private void Recruitment_Load(object sender, EventArgs e)
        {

        }

        private void metroLabel16_Click(object sender, EventArgs e)
        {

        }

        private void btnClear_Click(object sender, EventArgs e)
        {

        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtFirstName.Text != "")
            {
                string dt = DateTime_DOB.Text;
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Recruitment
(First_Name,Middle_Name,Last_Name,Gender,Date_Of_Birth,Email_ID,Phone
_No,Education,Specialization,Martial_Status,Experience,Former_Company
,Designation,Salary,Referred_By,Employee_ID,Employee_Name) VALUES('"
+ txtFirstName.Text + "','" + txtMiddleName.Text + "','" +
txtLastName.Text + "','" + comboGender.Text + "','" +
DateTime_DOB.Text + "','" + txtEmailID.Text + "','" + txtPhoneNo.Text
+ "','" + comboEducation.Text + "','" + comboSpecialization.Text +
"','" + comboMartialStatus.Text + "','" + comboExperience.Text +
"','" + txtFormerCompany.Text + "','" + txtDesignation.Text + "','" +
txtSalary.Text + "','" + comboReferredBy.Text + "','" + txtEmpID.Text
+ "','" + txtEmpName.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Your Data is Successfully Saved");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void btnUpdate_Click(object sender, EventArgs e)
        {

        }

        private void txtFirstName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtMiddleName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtLastName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtEmailID_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsEmailId(e);
        }

        private void txtPhoneNo_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtFormerCompany_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtDesignation_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtSalary_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtEmpID_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }
        private void txtEmpName_KeyPress(object sender, 
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }


    }
}
```

Human Resource Management with C#: Recruitment Details(Admin)

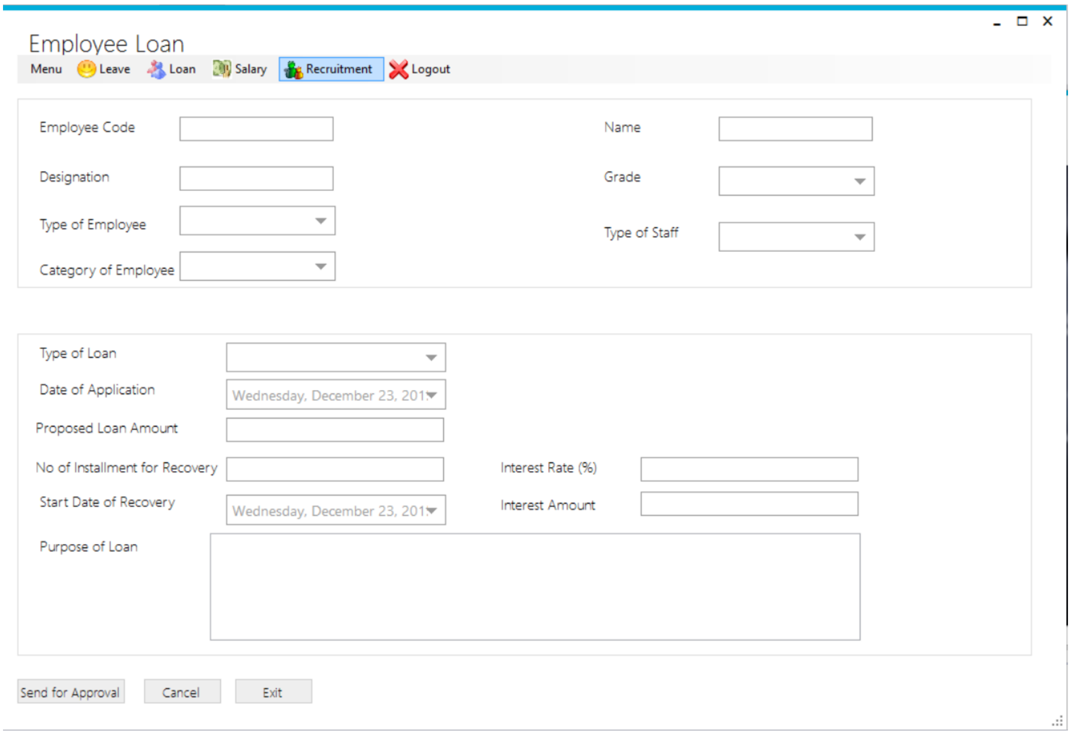


	Id	First_Name	Middle_Name	Last_Name	Gender	Date_OF_Birth	Email_ID	Phone_No	Education	Specialization
▸	1	Ramesh	kumar	Adhikari	Male	Wednesday, Jan...	ramesh2010@gm...	9806050903	Masters	Software
*										

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;
using ExcelLibrary.SpreadSheet;
using ExcelLibrary.BinaryFileFormat;
using ExcelLibrary.BinaryDrawingFormat;

namespace Human_Resourse_Management.Admin
{
    public partial class Recruitment_Details : MetroForm
    {
     Human_Resourse_Management.Database db = new 
Human_Resourse_Management.Database();
```

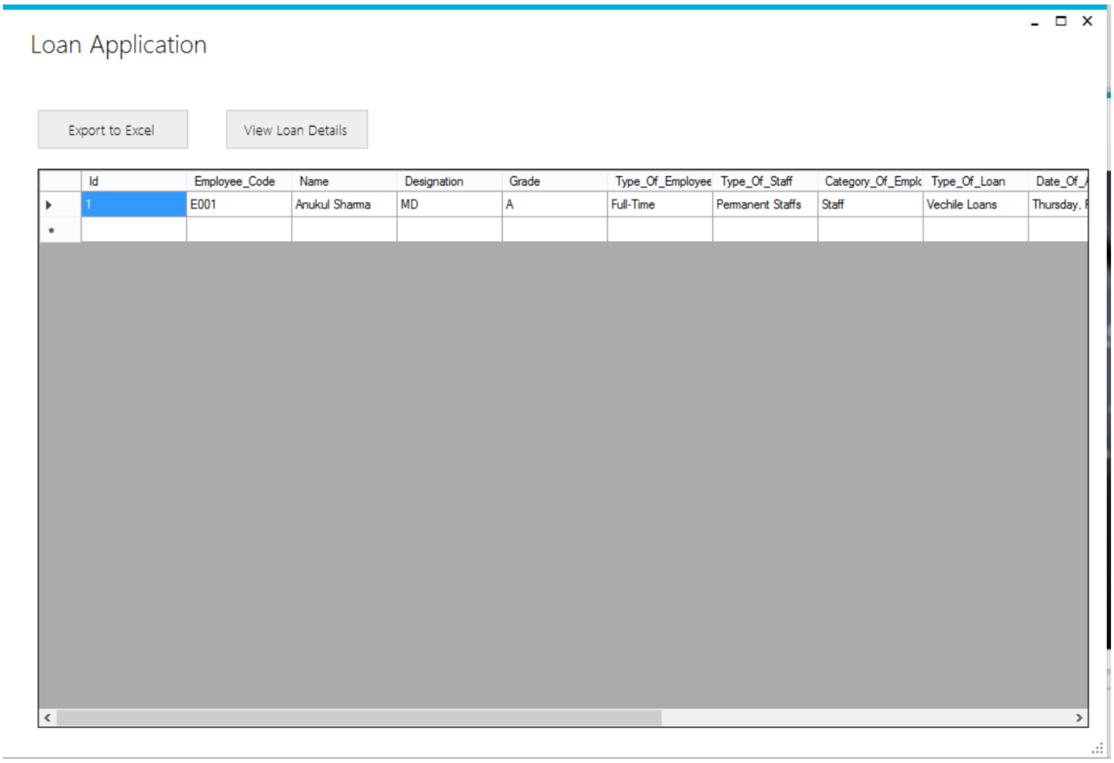
```
        public Recruitment_Details()
        {
            InitializeComponent();
        }

        private void Recruitment_Details_Load(object sender,
EventArgs e)
        {

        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnExportToExcel_Click(object sender, EventArgs
e)
        {
            // Creating a Excel object.
            Microsoft.Office.Interop.Excel._Application excel = new
Microsoft.Office.Interop.Excel.Application();
            Microsoft.Office.Interop.Excel._Workbook workbook =
excel.Workbooks.Add(Type.Missing);
            Microsoft.Office.Interop.Excel._Worksheet worksheet =
null;

            try
            {

                worksheet = workbook.ActiveSheet;

                worksheet.Name = "ExportedFromDatGrid";

                int cellRowIndex = 1;
                int cellColumnIndex = 1;

                //Loop through each row and read value from each
column.
                for (int i = 0; i < dataGridView1.Rows.Count - 1;
i++)
                {
                    for (int j = 0; j < dataGridView1.Columns.Count;
j++)
                    {
                        // Excel index starts from 1,1. As first Row
would have the Column headers, adding a condition check.
                        if (cellRowIndex == 1)
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = dataGridView1.Columns[j].HeaderText;
                        }
                        else
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = dataGridView1.Rows[i].Cells[j].Value.ToString();
                        }
                        cellColumnIndex++;
                    }
                    cellColumnIndex = 1;
                    cellRowIndex++;
                }
                //Getting the location and file name of the excel to
save from user.
                SaveFileDialog saveDialog = new SaveFileDialog();
                saveDialog.Filter = "Excel files (*.xlsx)|*.xlsx|All
files (*.*)|*.*";
                saveDialog.FilterIndex = 2;

                if (saveDialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
                {
                    workbook.SaveAs(saveDialog.FileName);
                    MessageBox.Show("Export Successful");
                }
            }
            catch (System.Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                excel.Quit();
                workbook = null;
                excel = null;
            }

        }

        private void btnView_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = @"Data Source=Anukul-
pc\sqlexpress;Initial Catalog=CHRMS;Integrated Security=True";
            SqlCommand command = new SqlCommand();
            command.Connection = con;
            command.CommandText = "SELECT * FROM  Recruitment";
            DataTable data = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            adapter.Fill(data);
            dataGridView1.DataSource = data;
        }
    }
}
```

Human Resource Management with C#: Employee Loan (Admin)



```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management.Employee
{
    public partial class Employee_Loan : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Employee_Loan()
        {
            InitializeComponent();
        }
         public Employee_Loan(int ID)
         {
            InitializeComponent();
            btnSend.Text = "Send";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT    *
FROM  Employee_Loan where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }

        }
        private void Employee_Loan_Load(object sender, EventArgs e)
        {

        }

        private void ComboGrade_SelectedIndexChanged(object sender,
EventArgs e)
        {

        }

        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            foreach (Control c in groupBox1.Controls)
            {
                c.Text = "";

            }
            foreach (Control b in groupBox2.Controls)
            {
                b.Text = "";
            }
        }

        private void btnSend_Click(object sender, EventArgs e)
        {
            if (txtName.Text != "")
            {
                string dt = DateTime_DateOfApplication.Text;
                string dt2 = DateTime_StartDateOfRecovery.Text;
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Employee_Loan
(Employee_Code,Name,Designation,Grade,Type_Of_Employee,Type_Of_Staff,
Category_Of_Employee,Type_Of_Loan,Date_Of_Application,Proposed_Loan_A
mount,No_Of_Installment_For_Recovery,Start_Date_Of_Recovery,Interest_
Rate,Interest_Amount,Purpose_Of_Loan) VALUES('" + txtEmpCode.Text +
"','" + txtName.Text + "','" + txtDesignation.Text + "','" +
ComboGrade.Text + "','" + comboTypeOfEmployee.Text + "','" +
comboTypeOfStaff.Text + "','" + comboCategoryOfEmployee.Text + "','"
+ comboTypeOfLoan.Text + "','" + DateTime_DateOfApplication.Text +
"','" + txtProposedAmount.Text + "','" + txtNoOfInstallment.Text +
"','" + DateTime_StartDateOfRecovery.Text + "','" +
txtInterestRate.Text + "','" + txtInterestAmount.Text + "','" +
textBoxPurposeOfLoan.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Message is Sent");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }
    }
}
```

Human Resource Management with C#: Loan Application (Admin)

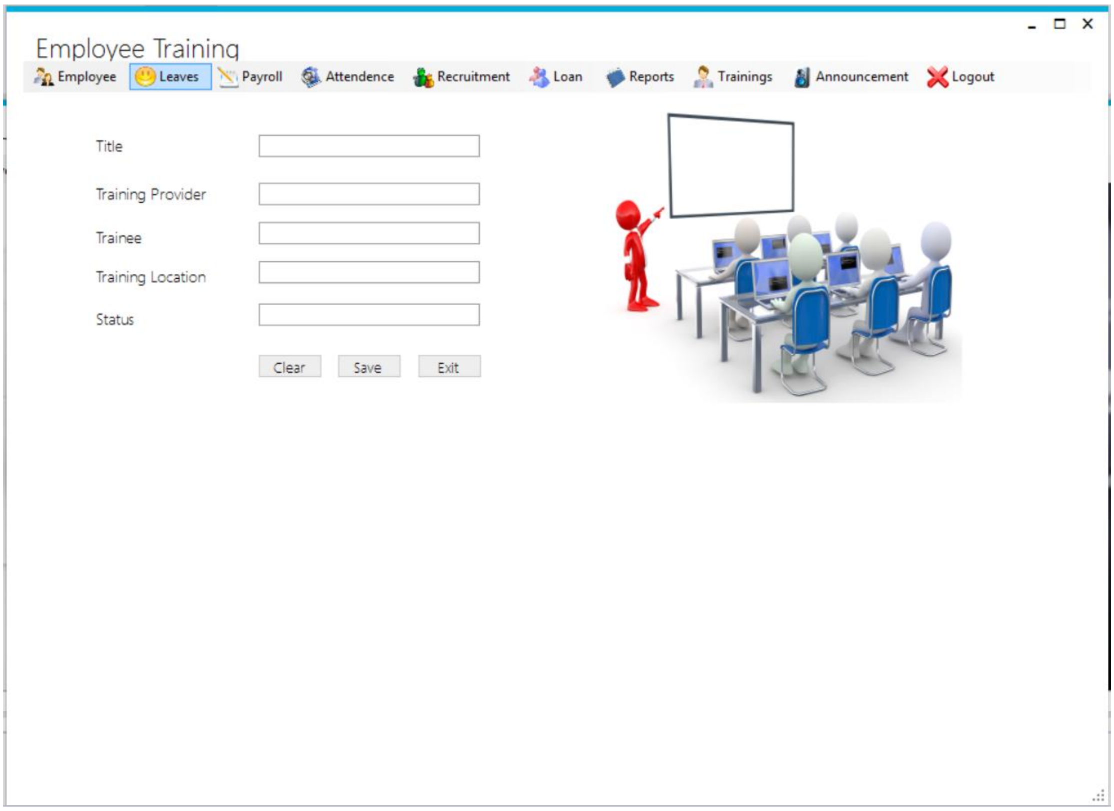


Loan Application

Id	Employee_Code	Name	Designation	Grade	Type_Of_Employee	Type_Of_Staff	Category_Of_Emplc	Type_Of_Loan	Date_Of_/
1	E001	Anukul Sharma	MD	A	Full-Time	Permanent Staffs	Staff	Vechile Loans	Thursday, F

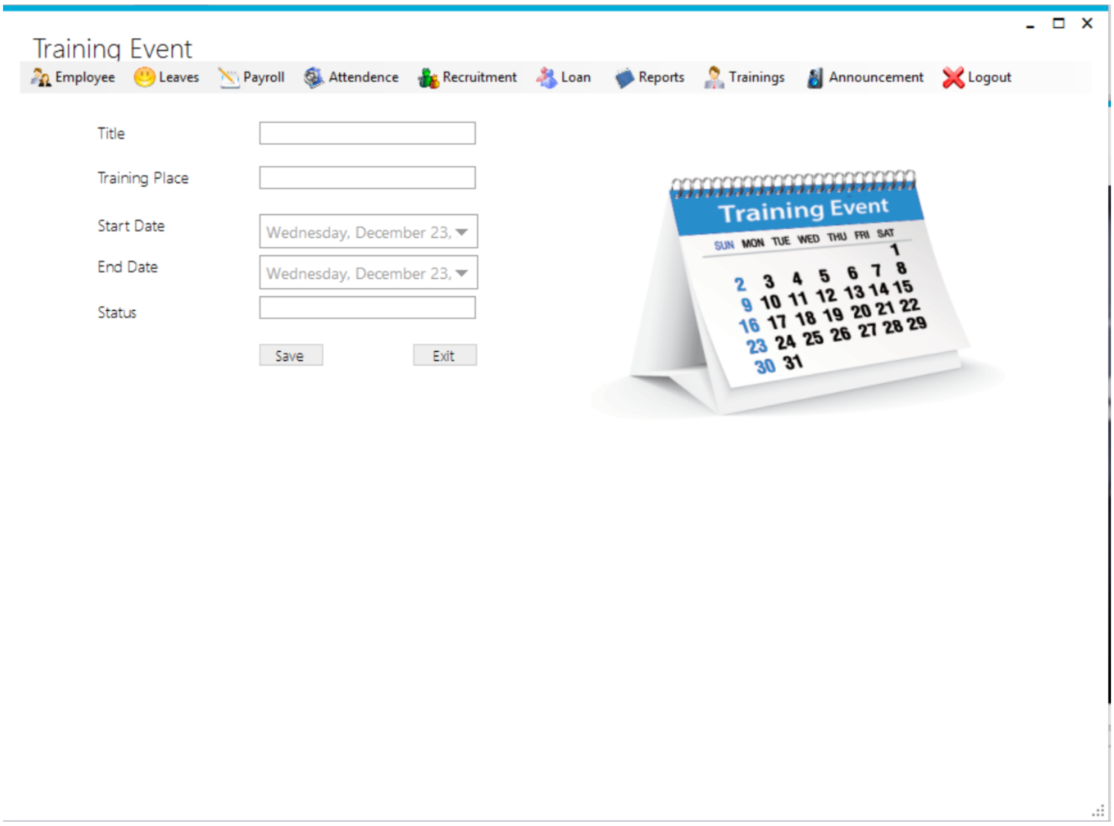
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MetroFramework.Forms;
using System.Data.SqlClient;
using ExcelLibrary.CompoundDocumentFormat;
using ExcelLibrary.SpreadSheet;
using ExcelLibrary.BinaryFileFormat;
using ExcelLibrary.BinaryDrawingFormat;

namespace Human_Resourse_Management.Admin
{
    public partial class btnExportToExcel : MetroForm
    {
        public btnExportToExcel()
        {
            InitializeComponent();
        }

        private void Loan_Application_Load(object sender, EventArgs
e)
        {

        }

        private void btnView_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = @"Data Source=Anukul-
pc\sqlexpress;Initial Catalog=CHRMS;Integrated Security=True";
            SqlCommand command = new SqlCommand();
            command.Connection = con;
            command.CommandText = "SELECT * FROM  Employee_Loan";
            DataTable data = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            adapter.Fill(data);
            dataGridView1.DataSource = data;

        }

        private void btnExport_Click(object sender, EventArgs e)
        {

            // Creating a Excel object.
            Microsoft.Office.Interop.Excel._Application excel = new
Microsoft.Office.Interop.Excel.Application();
            Microsoft.Office.Interop.Excel._Workbook workbook =
excel.Workbooks.Add(Type.Missing);
            Microsoft.Office.Interop.Excel._Worksheet worksheet =
null;

            try
            {

                worksheet = workbook.ActiveSheet;

                worksheet.Name = "ExportedFromDatGrid";

                int cellRowIndex = 1;
                int cellColumnIndex = 1;
                //Loop through each row and read value from each
column.
                for (int i = 0; i < dataGridView1.Rows.Count - 1;
i++)
                {
                    for (int j = 0; j < dataGridView1.Columns.Count;
j++)
                    {
                        // Excel index starts from 1,1. As first Row
would have the Column headers, adding a condition check.
                        if (cellRowIndex == 1)
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = dataGridView1.Columns[j].HeaderText;
                        }
                        else
                        {
                            worksheet.Cells[cellRowIndex,
cellColumnIndex] = dataGridView1.Rows[i].Cells[j].Value.ToString();
                        }
                        cellColumnIndex++;
                    }
                    cellColumnIndex = 1;
                    cellRowIndex++;
                }

                //Getting the location and file name of the excel to
save from user.
                SaveFileDialog saveDialog = new SaveFileDialog();
                saveDialog.Filter = "Excel files (*.xlsx)|*.xlsx|All
files (*.*)|*.*";
                saveDialog.FilterIndex = 2;

                if (saveDialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
                {
                    workbook.SaveAs(saveDialog.FileName);
                    MessageBox.Show("Export Successful");
                }
            }
            catch (System.Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                excel.Quit();
                workbook = null;
                excel = null;
            }

        }
    }
}
```

Human Resource Management with C#: Employee Training (Admin)



```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    public partial class Employee_Training : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Employee_Training()
        {
            InitializeComponent();
        }
        public Employee_Training(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Employee_Training  where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtTitle.Text = dt.Rows[0][0].ToString();
                    txtTrainingProvider.Text =
dt.Rows[0][1].ToString();
                }
            }
        }
        private void Employee_Training_Load(object sender, EventArgs
e)
        {

        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtTitle.Text != "")
            {
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Employee_Training
(Title,Training_Provider,Trainee,Training_Loaction,Status) VALUES('"
+ txtTitle.Text + "','" + txtTrainingProvider.Text + "','" +
txtTrainee.Text + "','" + txtTrainingLoaction.Text + "','" +
txtStatus.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Saved");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void btbClear_Click(object sender, EventArgs e)
        {
            foreach (Control c in Controls)
            {
                txtTitle.Text = "";
                txtTrainingProvider.Text = "";
                txtTrainee.Text = "";
                txtTrainingLoaction.Text = "";
                txtStatus.Text = "";

            }
        }
        private void logoutToolStripMenuItem_Click(object sender, 
EventArgs e)
        {
            this.Close();
        }
    }
}
```

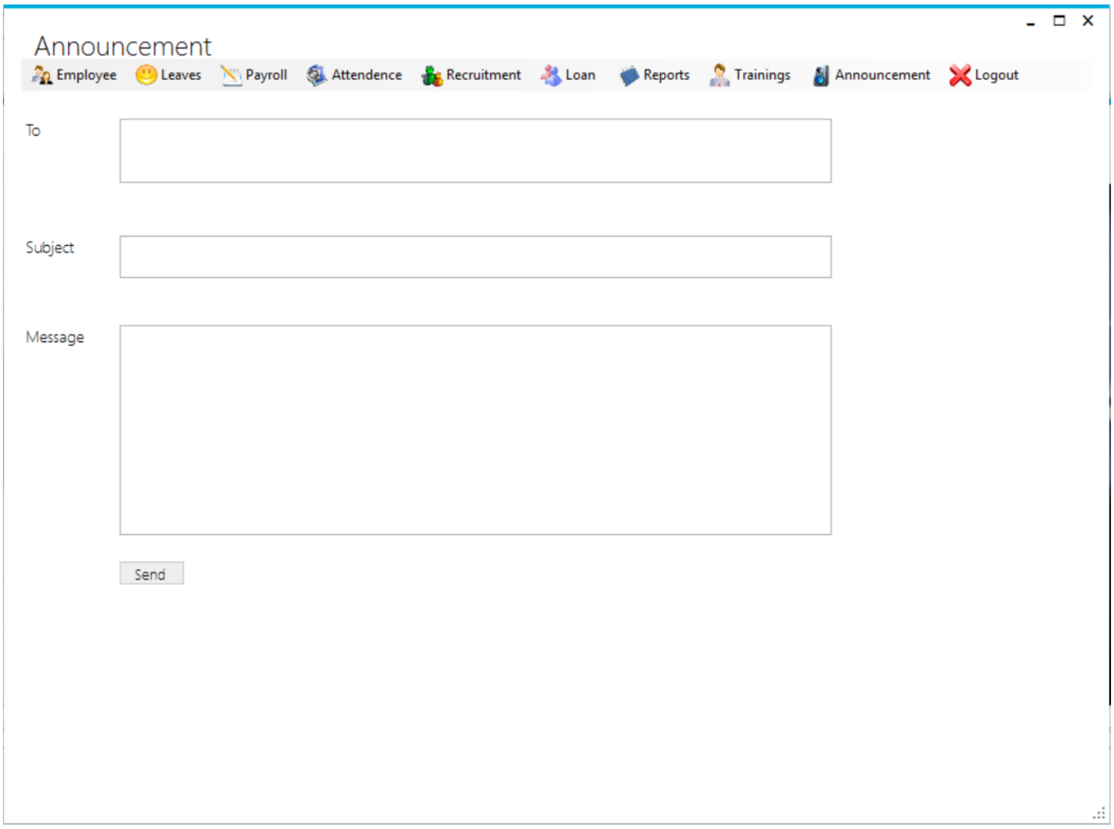
Human Resource Management with C#: Training Event (Admin)



```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    public partial class Training_Event : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Training_Event()
        {
            InitializeComponent();
        }
        public Training_Event(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Training_Event where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtTitle.Text = dt.Rows[0][0].ToString();
                    txtTrainingPlace.Text = dt.Rows[0][1].ToString();
                }
            }
        }
        private void Training_Event_Load(object sender, EventArgs e)
        {

        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {

            if (txtTitle.Text != "")
            {
                string dt = DateTime_StartDate.Text;
                string dt1 = DateTime_EndDate.Text;
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Training_Event  (Title,Training_Place,Start_Date_End_Date,Status)
VALUES('" + txtTitle.Text + "','" + txtTrainingPlace.Text + "','" +
DateTime_StartDate.Text + "','" + DateTime_EndDate.Text + "','" +
txtStatus.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Saved");


            }
        }

        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }
    }
```

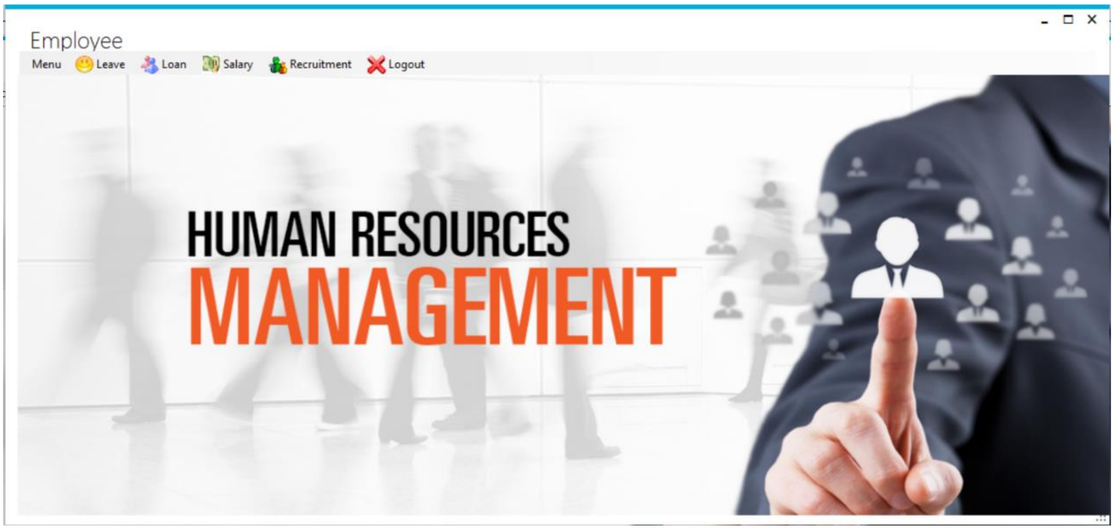
```
}
```

Human Resource Management with C#: Announcement (Admin)



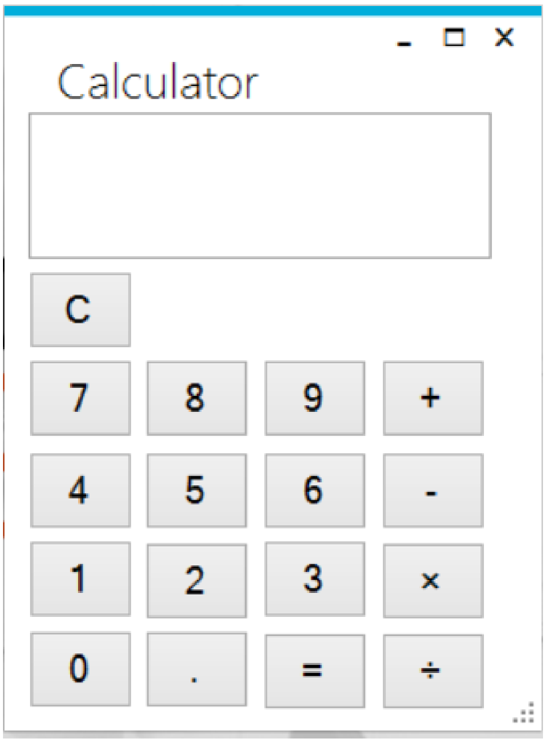
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    public partial class Announcement : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Announcement()
        {
            InitializeComponent();
        }
        public Announcement(int ID)
        {
            InitializeComponent();
            btnSend.Text = "Send";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Register where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    textBoxTo.Text = dt.Rows[0][0].ToString();
                    textBoxSubject.Text = dt.Rows[0][1].ToString();
                }
            }

        }
        private void Announcement_Load(object sender, EventArgs e)
        {

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }

        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }

        private void btnSend_Click(object sender, EventArgs e)
        {
            if (textBoxTo.Text != "")
            {
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Announcement (To,Subject,Message) VALUES('" + textBoxTo.Text + "','"
+ textBoxSubject.Text + "','" + textBoxMessage.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Saved");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }
    }
}
```

Human Resource Management with C#: Employee(Employee)



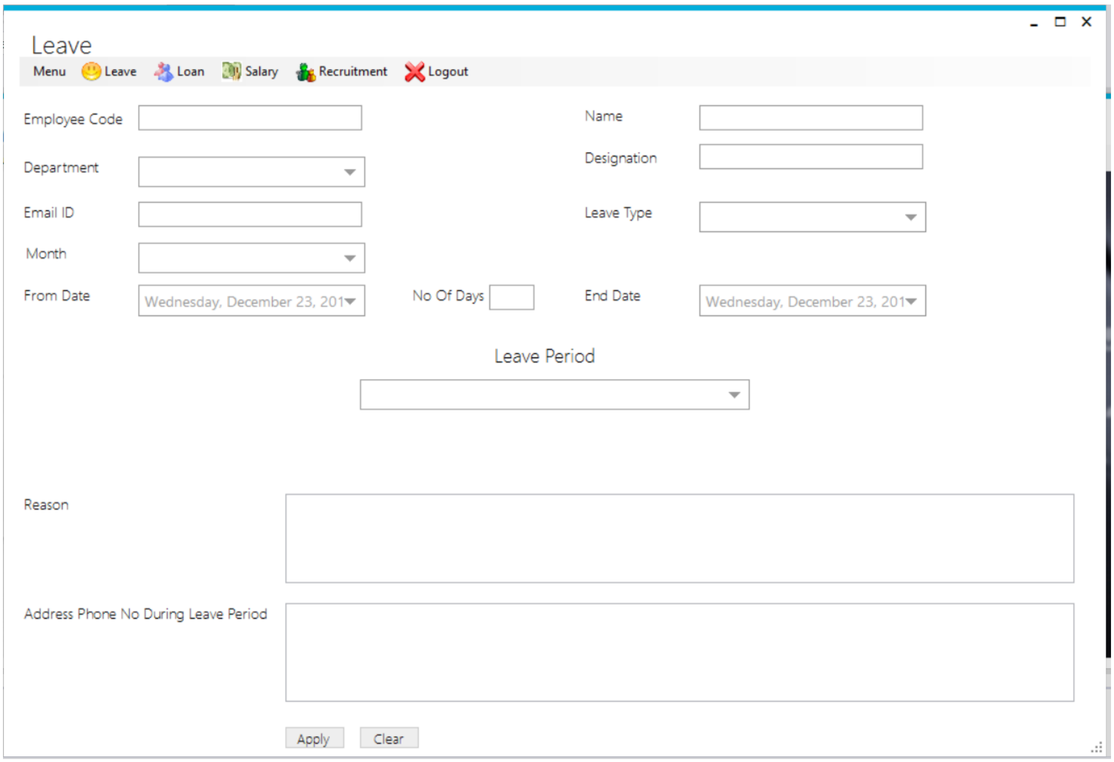
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MetroFramework.Forms;
using Human_Resourse_Management.Employee;

namespace Human_Resourse_Management
{
    public partial class SubMain : MetroForm
    {
        public SubMain()
        {
            InitializeComponent();
        }

        private void openToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            MessageBox.Show("you pressed open");
            new Start().ShowDialog();
            this.Hide();
        }

        private void menuStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {
        }

        private void SubMain_Load(object sender, EventArgs e)
        {
        }

        private void addNewToolStripMenuItem_Click(object sender,
EventArgs e)
        {

        }

        private void applyForLeavesToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new Employee_Leave().ShowDialog();
            this.Hide();
        }

        private void applyForLoanToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            new
Human_Resourse_Management.Employee.Employee_Loan().ShowDialog();
            this.Hide();
        }

        private void registerToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            new
Human_Resourse_Management.Admin.Recruitment().ShowDialog();
            this.Hide();
        }

        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }

        private void calculatorToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            new Human_Resourse_Management.Calculator().ShowDialog();
            this.Hide();
        }

        private void exitToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }

        private void SubMain_FormClosing(object sender,
FormClosingEventArgs e)
        {
            DialogResult dialog = MessageBox.Show("Do you Really want
to Exit the Program", "**********Human Resource Management
System*********", MessageBoxButtons.OKCancel);
            if (dialog == DialogResult.OK)
            {
                Application.Exit();
            }
            else if (dialog == DialogResult.Cancel)
            {
                e.Cancel = true;
            }
        }
    }
}
```

Human Resource Management with C#: Calculator (Employee)



```
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;
using MetroFramework.Forms;

namespace Human_Resourse_Management
{
    public partial class Calculator : MetroForm
    {
        private double accumulator = 0;
        private char lastOperation;

        public Calculator()
        {
            InitializeComponent();
        }

        private void Operator_Pressed(object sender, EventArgs e)
        {
            // An operator was pressed; perform the last operation
and store the new operator.
            char operation = (sender as Button).Text[0];
            if (operation == 'C')
            {
                accumulator = 0;
            }
            else
            {
                double currentValue = double.Parse(Display.Text);
                switch (lastOperation)
                {
                    case '+': accumulator += currentValue; break;
                    case '-': accumulator -= currentValue; break;
                    case '×': accumulator *= currentValue; break;
                    case '÷': accumulator /= currentValue; break;
                    default: accumulator = currentValue; break;
                }
            }

            lastOperation = operation;
            Display.Text = operation == '=' ? accumulator.ToString()
: "0";
        }

        private void Number_Pressed(object sender, EventArgs e)
        {
            // Add it to the display.
            string number = (sender as Button).Text;
            Display.Text = Display.Text == "0" ? number :
Display.Text + number;
        }

        private void Calculator_Load(object sender, EventArgs e)
        {

        }

    }
}
```

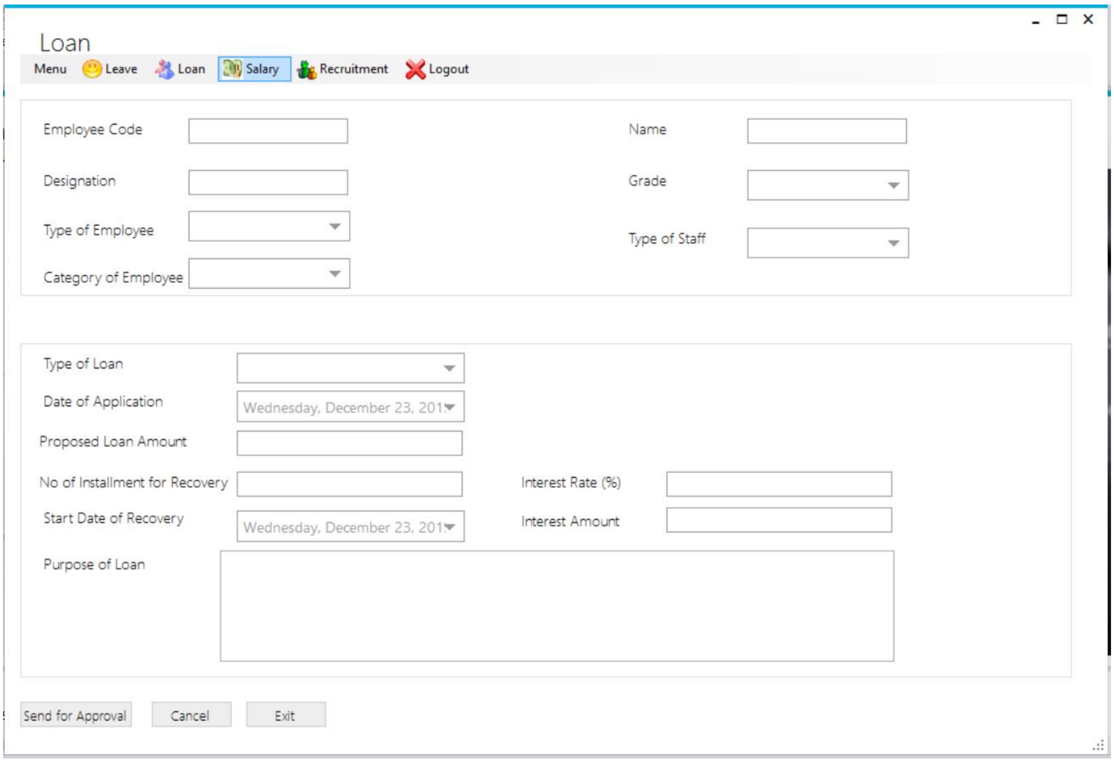
Human Resource Management with C#: Leave (Employee)



```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    public partial class Employee_Leave : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Employee_Leave()
        {
            InitializeComponent();
        }
        public Employee_Leave(int ID)
        {
            InitializeComponent();
            btnApply.Text = "Apply";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Employee_Leave where Name ='" + ID + "'", con);
```

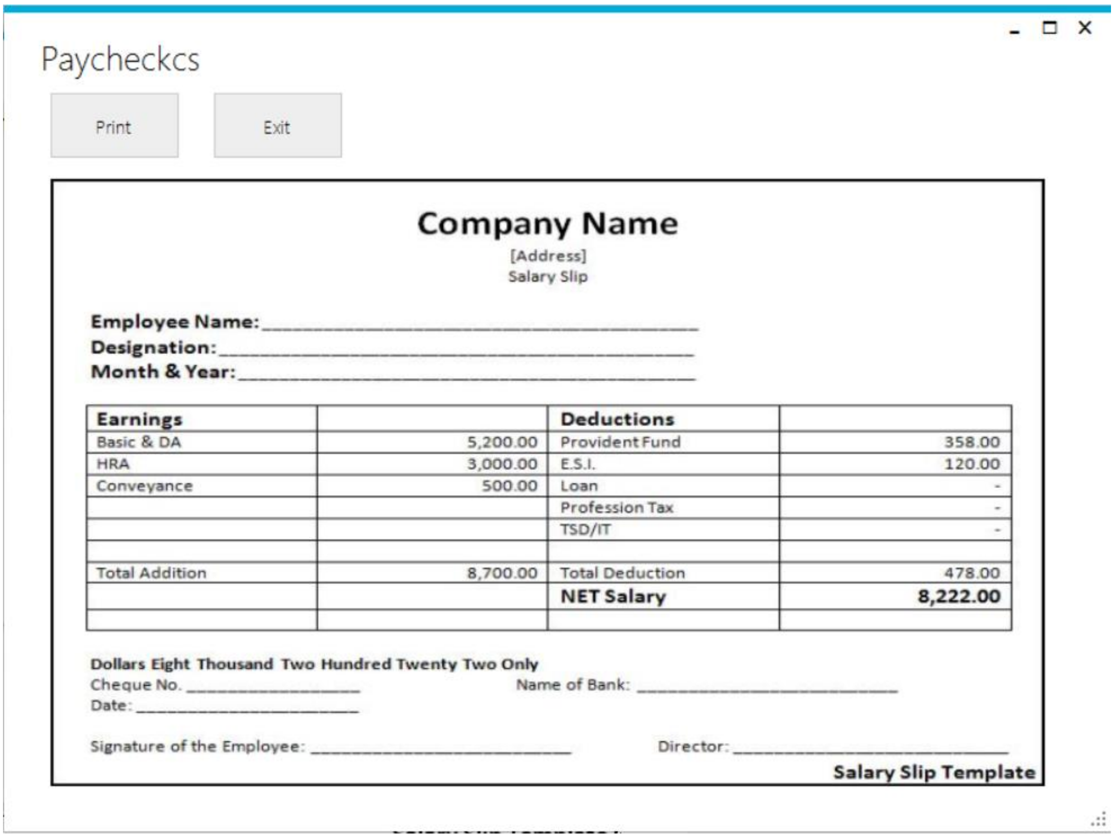
```
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }

        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }

        private void btnClear_Click(object sender, EventArgs e)
        {
            {
                foreach (Control b in Controls)
                {
                    txtEmpCode.Text = "";
                    txtName.Text = "";
                    txtDesignation.Text = "";
                    txtEmailID.Text = "";
                    txtNoOfDays.Text = "";
                    txtReason.Text = "";
                    txtAddress.Text = "";
                    comboDepartment.Text = "";
                    comboLeaveType.Text = "";
                    comboMonth.Text = "";
                    DateTime_EndDate.Text = "";
                    DateTime_FromDate.Text = "";
                    comboLeavePeriod.Text = "";
                }
            }
        }

        private void btnApply_Click(object sender, EventArgs e)
        {
            if (txtName.Text != "")
            {
                string ds1 = DateTime_FromDate.Text;
                string ds2 = DateTime_EndDate.Text;
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Employee_Leave
(Employee_Code,Name,Department,Designation,Email_ID,Leave_Type,Month,
No_Of_Days,Leave_Period,Reason,Address,From_Date,End_Date) VALUES('"
+ txtEmpCode.Text + "','" + txtName.Text + "','" +
comboDepartment.Text + "','" + txtDesignation.Text + "','" +
txtEmailID.Text + "','" + comboLeaveType.Text + "','" +
comboMonth.Text + "','" + txtNoOfDays.Text + "','" +
comboLeavePeriod.Text + "','" + txtReason.Text + "','" +
txtAddress.Text + "','" + DateTime_FromDate.Text + "','" +
DateTime_EndDate.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Data is Sent");
            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void Employee_Leave_Load(object sender, EventArgs e)
        {

        }

        private void txtEmpCode_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }


        private void txtName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtDesignation_KeyPress(object sender,
KeyPressEventArgs e)
        {

            Validation.IsAlpha(e);
        }

        private void txtEmailID_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsEmailId(e);
        }

        private void txtNoOfDays_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

    }
}
```

Human Resource Management with C#: Loan (Employee)



```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management.Employee
{
    public partial class Employee_Loan : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Employee_Loan()
        {
            InitializeComponent();
        }
         public Employee_Loan(int ID)
         {
            InitializeComponent();
            btnSend.Text = "Send";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Employee_Loan where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                    txtEmpCode.Text = dt.Rows[0][0].ToString();
                    txtName.Text = dt.Rows[0][1].ToString();
                }
            }

        }
        private void Employee_Loan_Load(object sender, EventArgs e)
        {

        }

        private void ComboGrade_SelectedIndexChanged(object sender,
EventArgs e)
        {

        }

        private void logoutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.Close();
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            foreach (Control c in groupBox1.Controls)
            {
                txtEmpCode.Text = "";
                txtName.Text = "";
                txtDesignation.Text = "";
                ComboGrade.Text = "";
                comboTypeOfEmployee.Text = "";
                comboTypeOfStaff.Text = "";
                comboCategoryOfEmployee.Text = "";


            }
            foreach (Control b in groupBox2.Controls)
            {
                comboTypeOfLoan.Text = "";
                DateTime_DateOfApplication.Text = "";
                txtProposedAmount.Text = "";
                txtNoOfInstallment.Text = "";
                txtInterestRate.Text = "";
                DateTime_StartDateOfRecovery.Text = "";
                txtInterestAmount.Text = "";
                textBoxPurposeOfLoan.Text = "";

            }
        }

        private void btnSend_Click(object sender, EventArgs e)
        {
            if (txtName.Text != "")
            {
                string dt = DateTime_DateOfApplication.Text;
                string dt2 = DateTime_StartDateOfRecovery.Text;
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Employee_Loan
(Employee_Code,Name,Designation,Grade,Type_Of_Employee,Type_Of_Staff,
Category_Of_Employee,Type_Of_Loan,Date_Of_Application,Proposed_Loan_A
mount,No_Of_Installment_For_Recovery,Start_Date_Of_Recovery,Interest_
Rate,Interest_Amount,Purpose_Of_Loan) VALUES('" + txtEmpCode.Text +
"','" + txtName.Text + "','" + txtDesignation.Text + "','" +
ComboGrade.Text + "','" + comboTypeOfEmployee.Text + "','" +
comboTypeOfStaff.Text + "','" + comboCategoryOfEmployee.Text + "','"
+ comboTypeOfLoan.Text + "','" + DateTime_DateOfApplication.Text +
"','" + txtProposedAmount.Text + "','" + txtNoOfInstallment.Text +
"','" + DateTime_StartDateOfRecovery.Text + "','" +
txtInterestRate.Text + "','" + txtInterestAmount.Text + "','" +
textBoxPurposeOfLoan.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Message is Sent");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void txtEmpCode_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }

        private void txtName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtDesignation_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtProposedAmount_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtNoOfInstallment_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }
        private void txtInterestRate_KeyPress(object sender, 
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtInterestAmount_KeyPress(object sender, 
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }
    }
}
```

Human Resource Management with C#: Paycheck (Employee)



Company Name
[Address]
Salary Slip

Employee Name:____
Designation:____
Month & Year:____

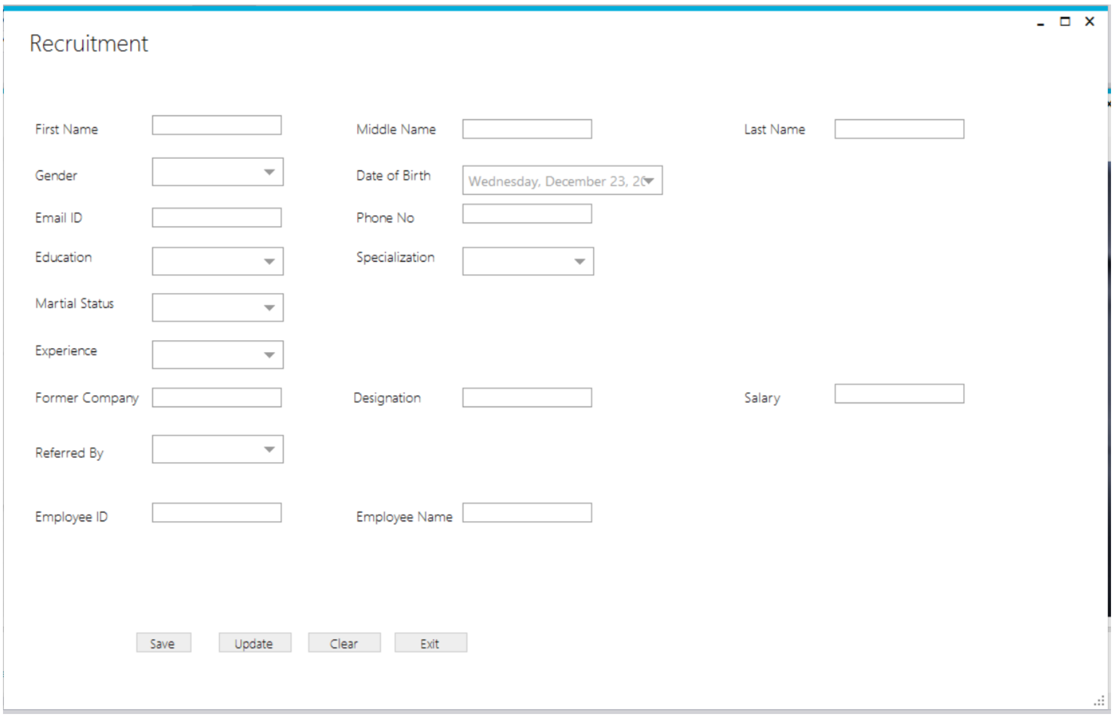
Earnings		**Deductions**	
Basic & DA	5,200.00	Provident Fund	358.00
HRA	3,000.00	E.S.I.	120.00
Conveyance	500.00	Loan	-
		Profession Tax	-
		TSD/IT	-
Total Addition	8,700.00	Total Deduction	478.00
		NET Salary	**8,222.00**

Dollars Eight Thousand Two Hundred Twenty Two Only
Cheque No. _________________ Name of Bank: _________________________
Date: ______________________

Signature of the Employee: _________________________ Director: ___________________________
Salary Slip Template

```
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;
using MetroFramework.Forms;
using System.Drawing.Printing;

namespace Human_Resourse_Management.Employee
{
    public partial class Paycheckcs : MetroForm
    {
        public Paycheckcs()
        {
            InitializeComponent();
        }

        private void Paycheckcs_Load(object sender, EventArgs e)
        {

        }
        private void myPrintDocument1_PrintPage(System.Object sender,
System.Drawing.Printing.PrintPageEventArgs e)
        {

            Bitmap myBitmap1 = new Bitmap(pictureBox1.Width,
pictureBox1.Height);

            pictureBox1.DrawToBitmap(myBitmap1, new Rectangle(0, 0,
pictureBox1.Width, pictureBox1.Height));

            e.Graphics.DrawImage(myBitmap1, 0, 0);

            myBitmap1.Dispose();

        }
        private void btnPrint_Click(object sender, EventArgs e)
        {

            System.Drawing.Printing.PrintDocument myPrintDocument1 =
new System.Drawing.Printing.PrintDocument();

            PrintDialog myPrinDialog1 = new PrintDialog();

            myPrintDocument1.PrintPage += new
System.Drawing.Printing.PrintPageEventHandler(myPrintDocument1_PrintP
age);

            myPrinDialog1.Document = myPrintDocument1;



            if (myPrinDialog1.ShowDialog() == DialogResult.OK)
            {

                myPrintDocument1.Print();

            }
        }

        private void btnClose_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Human Resource Management with C#: Recruitment (Employee)



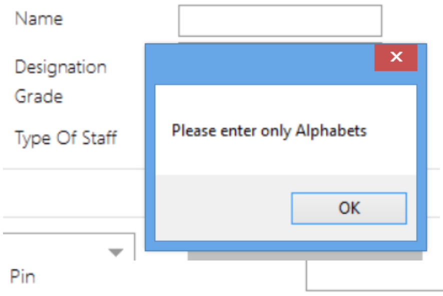
Wednesday, December 23, 20

```
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;
using MetroFramework.Forms;
using System.Data.SqlClient;

namespace Human_Resourse_Management.Admin
{
    public partial class Recruitment : MetroForm
    {
        Human_Resourse_Management.Database db = new
Human_Resourse_Management.Database();
        public Recruitment()
        {
            InitializeComponent();
        }
        public Recruitment(int ID)
        {
            InitializeComponent();
            btnSave.Text = "Save";
            {
                SqlConnection con = new SqlConnection(db.cs);
                SqlDataAdapter da = new SqlDataAdapter("SELECT   *
FROM  Register where Name ='" + ID + "'", con);
                DataTable dt = new DataTable();
                int i = da.Fill(dt);
                if (i > 0)
                {
                  txtFirstName.Text = dt.Rows[0][0].ToString();
                    txtMiddleName.Text = dt.Rows[0][1].ToString();
                }
            }

        }

        private void Recruitment_Load(object sender, EventArgs e)
        {

        }

        private void metroLabel16_Click(object sender, EventArgs e)
        {

        }

        private void btnClear_Click(object sender, EventArgs e)
        {

        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtFirstName.Text != "")
            {
                string dt = DateTime_DOB.Text;
                SqlConnection con = new
SqlConnection(db.connectionString);
                SqlCommand cmd = new SqlCommand("INSERT INTO
Recruitment
(First_Name,Middle_Name,Last_Name,Gender,Date_Of_Birth,Email_ID,Phone
_No,Education,Specialization,Martial_Status,Experience,Former_Company
,Designation,Salary,Referred_By,Employee_ID,Employee_Name) VALUES('"
+ txtFirstName.Text + "','" + txtMiddleName.Text + "','" +
txtLastName.Text + "','" + comboGender.Text + "','" +
DateTime_DOB.Text + "','" + txtEmailID.Text + "','" + txtPhoneNo.Text
+ "','" + comboEducation.Text + "','" + comboSpecialization.Text +
"','" + comboMartialStatus.Text + "','" + comboExperience.Text +
"','" + txtFormerCompany.Text + "','" + txtDesignation.Text + "','" +
txtSalary.Text + "','" + comboReferredBy.Text + "','" + txtEmpID.Text
+ "','" + txtEmpName.Text + "')", con);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Your Data is Successfully Saved");


            }
            else
            {
                MessageBox.Show("Enter All the Fields");
            }
        }

        private void btnUpdate_Click(object sender, EventArgs e)
        {

        }

        private void txtFirstName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtMiddleName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtLastName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtEmailID_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsEmailId(e);
        }

        private void txtPhoneNo_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtFormerCompany_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtDesignation_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }

        private void txtSalary_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsInteger(e);
        }

        private void txtEmpID_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlphaNumeric(e);
        }
        private void txtEmpName_KeyPress(object sender,
KeyPressEventArgs e)
        {
            Validation.IsAlpha(e);
        }


    }
}
```

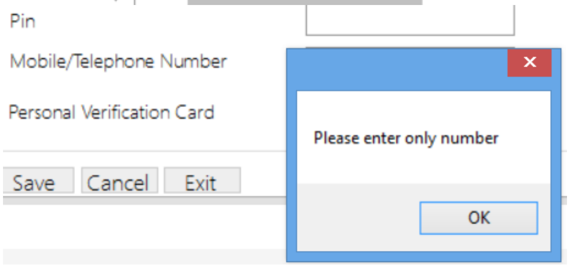
Human Resource Management with C#: Database Class (For Database Connection)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading.Tasks;
using System.Data.SqlClient;

namespace Human_Resourse_Management
{
    class Database
    {
        public String connectionString = "";
        public String cs="Data Source=.\\sqlexpress;Initial
Catalog=CHRMS;Integrated Security=True";

        public Database()
    {
       this.connectionString = cs;

    }
        public Database(String connectionString)
        {
            this.connectionString = connectionString;
        }
        public Database(String server, String database)
        {
            this.connectionString =
string.Format("Server={0};Database={1}; Trusted_Connection=True;",
server, database);
        }
        public Database(String server, String database, String user,
String pass)
        {

this.connectionString=String.Format("Server={0};Database={1}; User
ID={2}; Password={3}; Trusted_Connection=False",
server,database,user,pass);
        }
         public virtual string ConnectionString
        {
            get
            {
                return this.connectionString;
            }
            set
            {
                this.connectionString = value;
            }
        }

        public System.Data.DataSet EXECUTE_FOR_REPORT(string CMD)
        {
            System.Data.SqlClient.SqlConnection con;
            con = new
System.Data.SqlClient.SqlConnection(this.ConnectionString);
            con.Open();
            SqlDataAdapter adp = new SqlDataAdapter(CMD, con);
            DataSet DS = new DataSet();

            adp.Fill(DS);
            con.Close();
            return DS;
        }

        }

  }
```

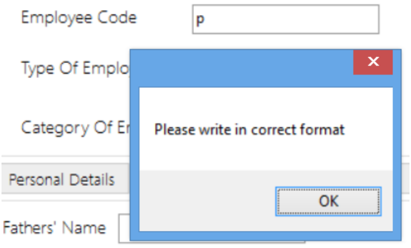
Human Resource Management with C#: Validation Class (Form Validation)



Please enter only Alphabets



Please enter only number



Please write in correct format

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using MetroFramework.Forms;

namespace Human_Resourse_Management
{
    class Validation
    {
        public static void IsAlpha(KeyPressEventArgs e)
        {
            string strToCheck = Convert.ToString(e.KeyChar);
            Regex ObjAlphaPattern = new Regex("^[a-zA-z]|[\b]|[ \t
]|[.]");
            if (!ObjAlphaPattern.IsMatch(strToCheck))
            {

                MessageBox.Show("Please enter only Alphabets");
                e.Handled = true;

            }

        }
        /****************************(EMAIL
VALIDATION)****************/

        public static void IsEmailId(KeyPressEventArgs e)
        {
            string strToCheck = Convert.ToString(e.KeyChar);
             Regex ObjAlphaPattern = new Regex("^[a-zA-Z0-
9]|[\b]|[\t]|[.]|[@]");
             if (!ObjAlphaPattern.IsMatch(strToCheck))
             {

               MessageBox.Show("Invalid Email Id");
            e.Handled = true;

             }
        }

        public static void checkEmail(string email,KeyPressEventArgs
e)
        {
           // check email
        }


        public static void IsInteger(KeyPressEventArgs e)
        {
            string strToCheck = Convert.ToString(e.KeyChar);
            Regex ObjIntPattern = new Regex("^[0-9]|[\b]|[ \t]|[+]");
            if (!ObjIntPattern.IsMatch(strToCheck))
            {

                MessageBox.Show("Please enter only number");
                e.Handled = true;
            }
        }
        public static void IsAlphaNumeric(KeyPressEventArgs e)
        {
            string strToCheck = Convert.ToString(e.KeyChar);
            Regex ObjAlphaPattern = new Regex("^[a-zA-z0-9]|[\b]|[
\t]");
            if (!ObjAlphaPattern.IsMatch(strToCheck))
            {

                MessageBox.Show("Please write in correct format");
                e.Handled = true;

            }
        }



        public static void IsQuantity(KeyPressEventArgs e)
        {
            string strToCheck = Convert.ToString(e.KeyChar);
            Regex ObjAlphaPattern = new Regex("^[0-9]|[.]|[\b]");
            if (!ObjAlphaPattern.IsMatch(strToCheck))
            {

                MessageBox.Show("Please write in correct format");
                e.Handled = true;

            }
        }
    }
}
```

Human Resource Management with C# Test Plan

Introduction

This document describes the user acceptance test plan for the Human Resource Management with C#. The complete test strategy for the Human Resource Management with C# is to perform the following kinds of tests, in sequence:

1. **Component testing** of each component that makes up the Human Resource Management with C#.
2. **Integration testing** of the Human Resource Management with C#, to ensure the correct interworking of its components.
3. **Validation testing** of the Human Resource Management with C#, to ensure that it works correctly in a pseudo-live environment.
4. **User acceptance testing** of the Human Resource Management with C#, to ensure that its function is acceptable to its users.

Acceptance testing is the last set of tests to be performed before the application goes officially live.

Test Strategy

The basis of user acceptance testing is that other tests were completed successfully, so the application and its required infrastructure are considered to be stable and reliable. Acceptance testing concentrates on the application from the user's perspective, that is, how the application is used and whether it meets the necessary quality criteria.

Change requests will be sent to the development team as the actionable documentation. Change criteria will be determined by the Test team and the Development team prior to the beginning of testing. For instance, criteria may include impact to desired functionality, amount of code impacted by proposed change, and design required by proposed change. The tester will evaluate the criteria. The test lead will determine Change Required or not. Once a bug has been determined as Change Required, the bug report will be translated into a Change Request and passed on to development.
The progress of the acceptance testing will be reported to the Administrator, together with any issues that are discovered and their planned resolutions. Log Out of the tests, and therefore the acceptance of the application, will be performed by the Administrator or a selected representative.

Preconditions

The following items are required before testing can take place:

- A complete and coherent functional specification of the Human Resource Management with C# expressed as usage scenarios.
- A complete and validation-tested release of the Human Resource Management with C#, delivered according to the delivery plan.
- An agreed-upon procedure for dealing with any anomalies that are discovered during the testing process.

- A set of test specifications describing how each functional area of the the Human Resource Management with C# is to be acceptance tested.
- An implemented test environment for the testing.
- Sufficient, suitable resources to carry out the testing.
- Available standards for the acceptance testing.

Test Priorities

During testing of the Human Resourse Management System, the following qualities will be tested in order of priority:

- Functionality—whether the required functions are available and working as expected.
- Usability—how user-friendly and intuitive the Human Resource Management with C# is.
- Security—how well-protected and guaranteed Admin and Employee data is.
- Performance—whether the response times are within acceptable limits
- Customization—how straightforward it is to use the application in new, unpredicted ways

Test Techniques

The following techniques will be applied:

- Scripted tests—sequences of user interactions (based on the usage scenarios) using predefined data sets against predicted results
- Unscripted tests—based on scripted tests, the tester tries to modify the scenarios to explore what-if possibilities
- Penetration tests—scripted tests to attempt unauthorized entry into the system
- Usability checklists—tests to determine the complexity of interactions
- Performance statistics—generation of performance information to check against desired performance criteria

Test Organization

Roles and Responsibilities

The following roles are defined:

- QA lead/test manager—responsible for planning and ensuring the smooth running of the test process.
- Tester—carries out the tests according to the test plan, and then reports the results.
- Product manager—ensures that the tests are carried out successfully from a user perspective.
- Project sponsor/client—acts as main stakeholder, and ensures that the needs of the customer community as a whole are considered.
- Test support—provides technical assistance, such as test environment configuration, and non-technical assistance, such as methodological support.

Weekly team meetings will be held involving the test manager, testers, and product managers. At these meetings, the progress of the testing process will be reported, any issues will be discussed, and actions will be agreed upon.

Deliverables

The following deliverables will be expected from the user acceptance testing process:

- Test plan—this document, together with any updates that have occurred during the testing process
- Change requests—any bugs, defects, or other changes required to the Human Resource Management System as a result of the testing process
- Weekly reports—progress reports to enable the status of the testing process to be determined
- Completion report—a report to be signed off by the customer, to signify the successful completion of the user acceptance testing

Test Environment

Hardware and Software

The test environment will consist of:

Hardware Requirements: Processor -1 GHz, RAM- 512, Disk Space (Minimum) - 32 bit (850MB) and 64 bit (1GB).

Software Requirements : Operating Systems – Windows (for better performance higher versions of windows are used)

Software – Visual Studio 2013 and Database &Microsoft SQL server 2012 and above.

Testing Automation Software

No testing automation software packages are selected at present.

Application Configuration

The following user accounts will be configured on the Form Itself.

Test Management

Tests shall be managed according to the corporate test management standards, which cover:

- Conduct of tests
- Reporting of test results
- Defect tracking and resolution
- Configuration management of the test environment
- Configuration control of test deliverables.

Testing Schedules

The user acceptance testing schedules are shown in the project structure document and resulting Gantt charts.

Threats to Testing

Potential threats to the testing process are as follows:

- **Insufficient resources available for testing.** Testing resources have been seconded from the development departments, whose time is at a premium. Mitigation: ensure department heads apply a high priority to the testing of Human Resource Management with C#.
- **Availability of sales personnel for testing.** The test team should be overseen by at least one sales representative. Mitigation: gain prior agreement from the vice president of Sales for two sales representatives to be assigned to test the application.

Conclusion

This is my first step towards my experience in developing this project and I personally feel that University's decision over this project is a good step in making students aware of the environment they are living and also the responsibilities they will be taking in the near future.

As matter of fact I learnt many new things that are not included in the syllabus like the way to do customer requirements analysis, designing an information system, steps that should be followed in software development etc. After doing this project I realize that before designing any information system, it is essential to understand user's requirements. One of the most crucial parts of doing any project is requirement analysis and system designing. It is because wrong analysis leads to wrong designing which ultimately results in wrong product. Testing is another step in software development that we cannot escape and it is very important to choose a good testing methods and procedures in order to develop a quality software package.

I also wanted to add more features but because of less time span I could not do so I would like to have a plan in doing in future. Last I would like to once again thank all my College, Teachers, Family and Friends for supporting and contributing in completing my project.

Anukul Sharma
1302007237

Bibliography

Websites

- http://www.google.com
- http://www.youtube.com
- http://www.microsoft.com
- http://www.programmer2programmer.net
- http://www.codeproject.com
- http://www.msdn.com
- http://www.tutorialspoint.com/csharp/csharp_overview.htm
- http://www.sqltuner.com
- http://www.stackoverflow.com
- http://thielj.github.io/MetroFramework

Books

- Microsoft Press - Microsoft Visual C# 2010 Step by Step
- Wrox - Beginning Visual C# 2010
- O'Reilly - SQL in a Nutshell, Second Edition
- Database Development in Visual C#
- Sams - Microsoft SQL Server High Availability

Training Videos

- Lynda.com C# Essential Training
- Lynda.com - Visual Studio 2010 Essential Training