SALARY MANAGEMENT SYSTEM DATABASE PROJECT

Final Project Report

Course: CS 4600 / 5200 Database Theory and Applications

CRN 13268

TEAM MEMBERS

TECHNICAL LEADER -SAI TEJASWI MANDOJI – 700758342

DATABASE DEVELOPER -SURYABHAVANA ATMAKURI – 700741469

DATABASE DEVELOPER -REENA REDDY JAKKA - 700741594

FRONTEND DEVELOPER -RAMYA SREE GUDAVALLI – 700759183

FRONTEND DEVELOPER -SAI VAMSHI THUMU – 700759298

FRONTEND DEVELOPER -HARI KRISHNA YAKKANTI – 700756159

ABSTRACT

Nowadays, a salary management system that is designed to automate and streamline the process of managing employee salaries and related financial transactions. The system provides a comprehensive set of tools and features to help companies manage employee payroll, including tracking time and attendance, calculating payroll taxes and deductions, and generating payslips and other financial reports. With this system, companies can ensure accurate and timely payments to employees, reduce errors and inefficiencies, and maintain compliance with legal and regulatory requirements. The system also provides managers with valuable insights into employee compensation trends, enabling them to make informed decisions regarding salary adjustments and other compensation-related matters. Overall, the salary management system is an essential tool for any company looking to manage their payroll processes more effectively and efficiently.

INDEX

1. Introduction:

A salary management system is a software application that helps companies manage and process employee salaries and related financial transactions. The system is designed to streamline payroll processes and ensure accurate and timely payments to employees. With a salary management system, companies can maintain records of employee salaries, track time and attendance, and automate payroll calculations, tax deductions, and other financial transactions. This allows companies to improve efficiency, reduce errors, and ensure compliance with legal and regulatory requirements. In addition, salary management systems provide managers with valuable insights into employee compensation trends and help them make informed decisions regarding salary adjustments and other compensation-related matters. There are five modules, and they are as follows,

- Employee Management
- Allowance and Deduction
- Salary Module
- User Account

SOFTWARE REQUIREMENTS:

Platform Support: Windows 7 or advanced.

Language: Windows forms, C#

Software: Visual Studio 2017

Database: SQL Server

2.Lessons Learned:

While implementing the project we learned a lot of things

1. How to gather requirements for the system to implement?

3. Creating and integrating databases.

4. Storing data in the database and usage of sessions.

5. Dealing with database failures and error handling

6. Handling issues of data inconsistency and memory allocation.

7. Acquired knowledge on event handling and code optimization techniques while implementing code behind.

3. Database Design Process

The database design process for a salary management database typically involves the following steps:

1. Identify the entities: Start by identifying the main entities in the salary management system.

2. Determine the relationships: Determine the relationships between the entities.

3. Define the attributes: Define the attributes for each entity, such as name, address, salary, and job title.

4. Normalize the data: Normalize the data to reduce redundancy and improve data integrity. This involves breaking down the data into smaller, more manageable tables.

5. Create a data model: Create a data model that shows the relationships between the entities and their attributes. This can be done using a data modeling tool or by creating an entity-relationship diagram (ERD).

6. Implement the database: Implement the database by creating tables, defining relationships, and setting up constraints to ensure data integrity.

7. Populate the database: Populate the database with data, such as employee information and salary details.

8. Test the database: Test the database to ensure that it works as expected and that the data is accurate.

9. Maintain the database: Finally, maintain the database by performing regular backups, updating data, and ensuring data security.

By following these steps, you can create a robust and efficient salary management database that meets the needs of your organization.

4. Database Structure:

A database structure refers to the organization and arrangement of data in a database system. A well-designed database structure is crucial for efficient data retrieval, storage, and management. Here are the main components of a typical database structure:

1. Tables: A table is the basic building block of a database structure. It contains rows and columns that store data. Each table represents a specific type of entity, such as customers, products, or orders.

2. Fields/Columns: Fields or columns are the individual data items within a table. They represent the specific attributes of an entity, such as name, address, or date of birth.

3. Primary Key: A primary key is a unique identifier for each row in a table. It ensures that each row can be uniquely identified and distinguishes it from other rows in the same table.

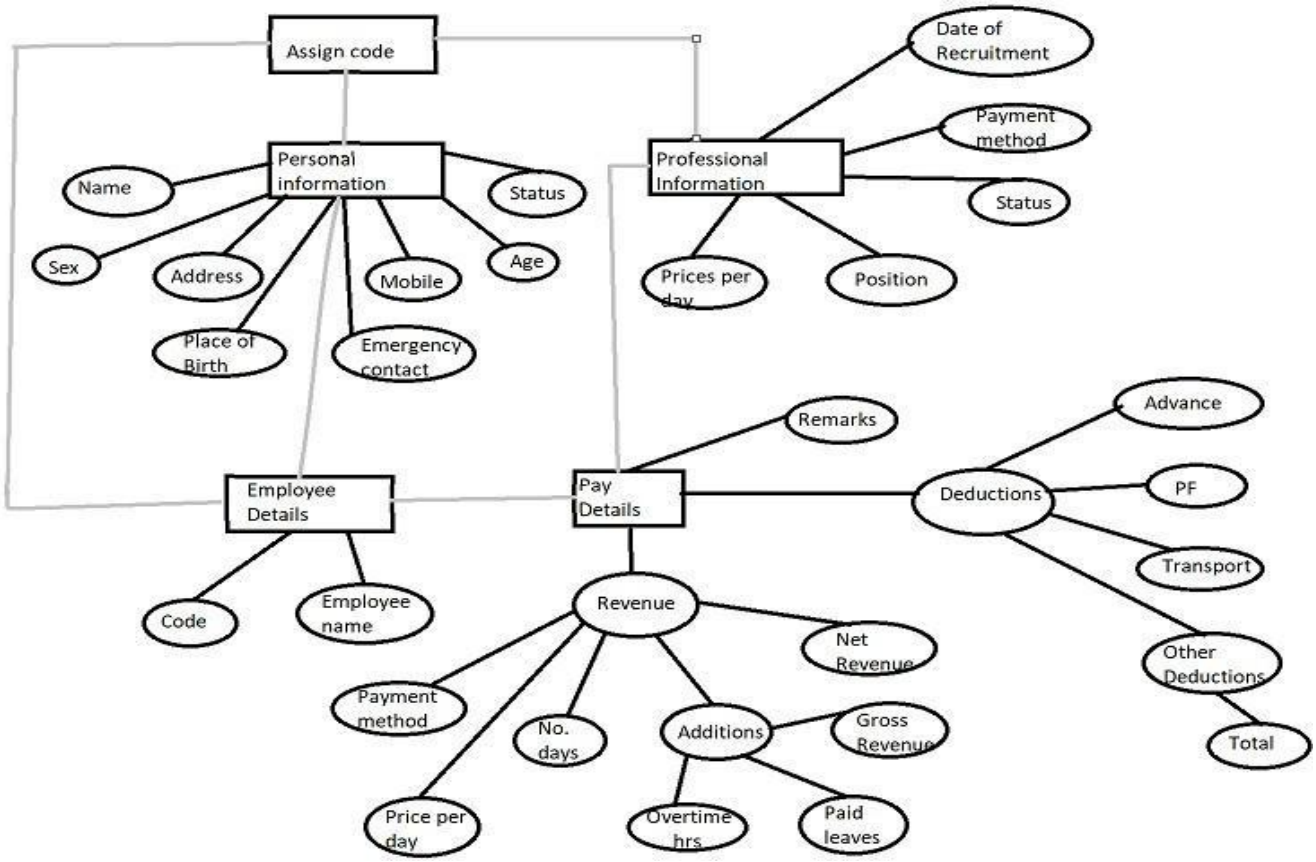
4. Foreign Key: A foreign key is a field in one table that refers to the primary key in another table. It establishes a relationship between the two tables and allows data to be linked between them.

5. Indexes: An index is a data structure that improves the performance of database queries by providing faster access to data. It is created on one or more columns in a table and allows for quick searches and retrievals of specific data.

6. Constraints: Constraints are rules that ensure data integrity and consistency. They are used to enforce business rules, prevent data duplication, and maintain data quality. Examples of constraints include primary key constraints, foreign key constraints, and check constraints.

By using these components, database designers can create a well- structured database that is efficient, scalable, and easy to maintain.

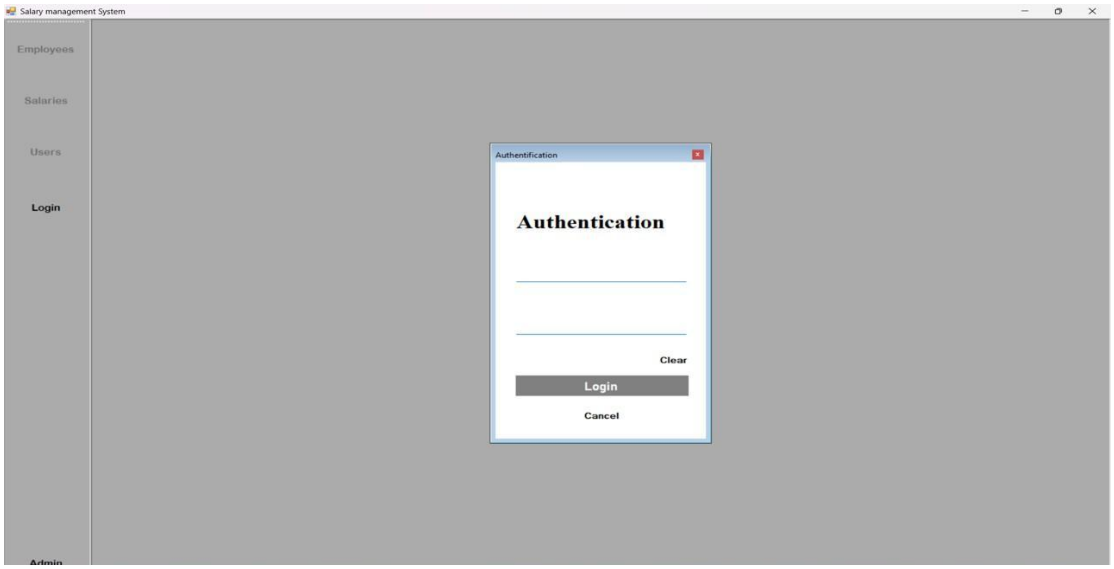
5. ER Diagram:



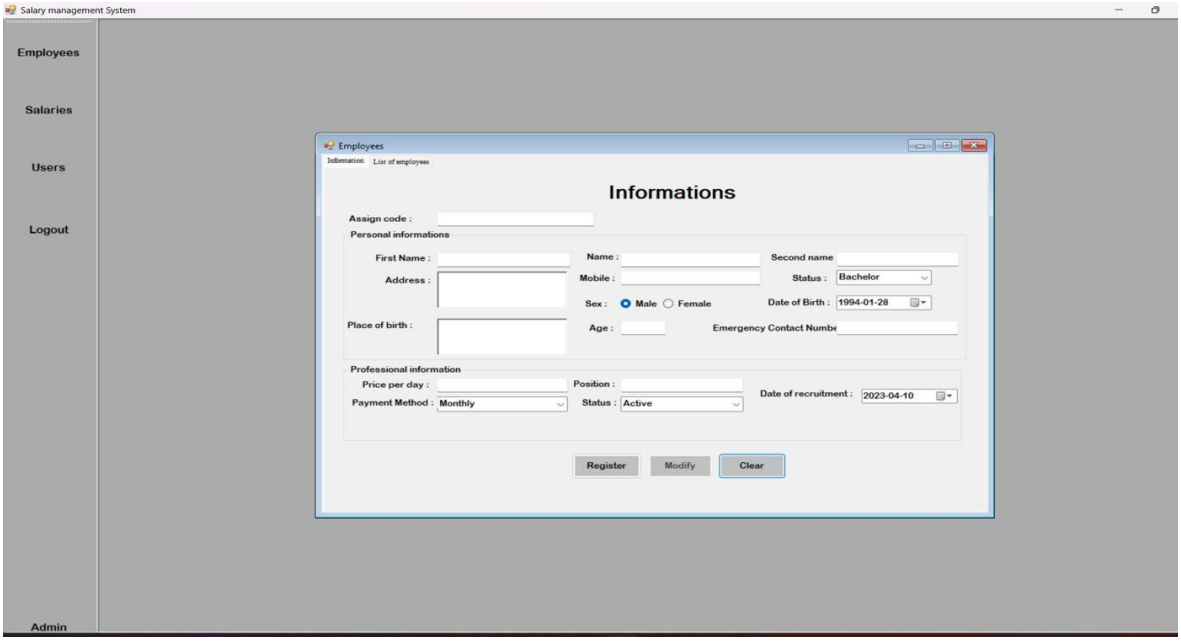
6. Project

Description: Login

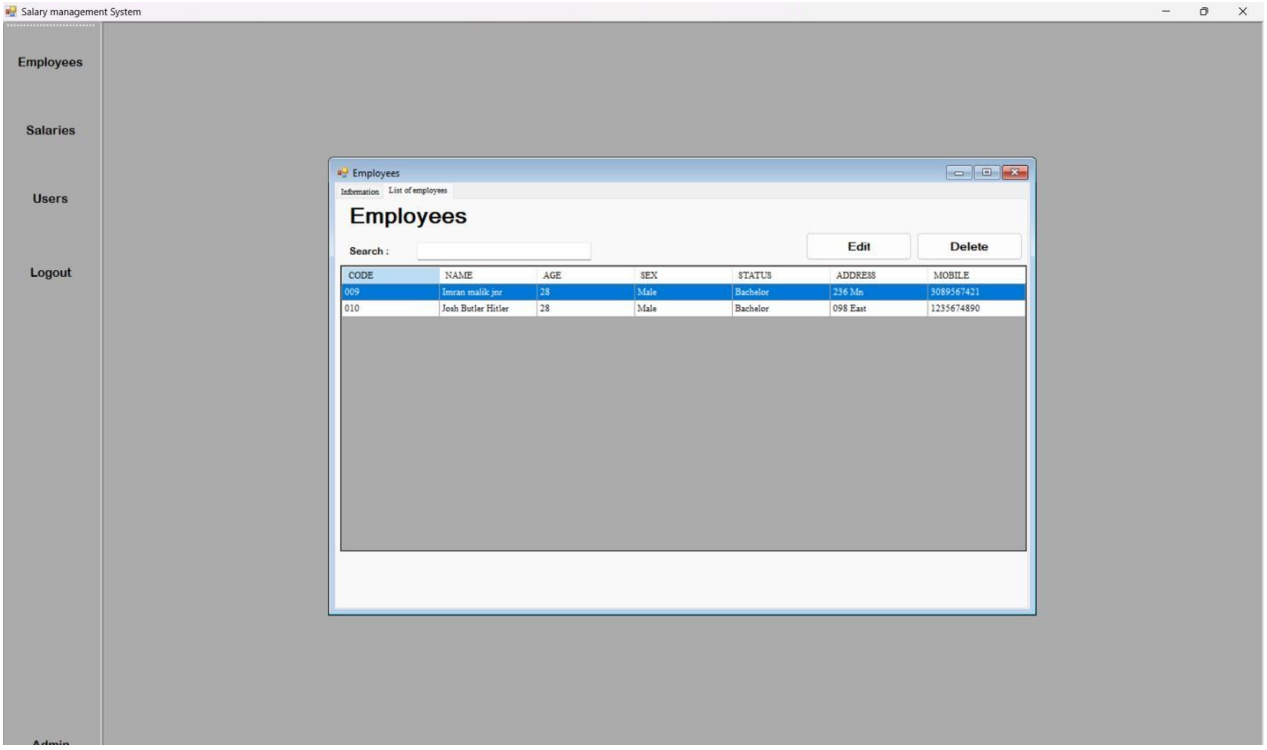
Page:



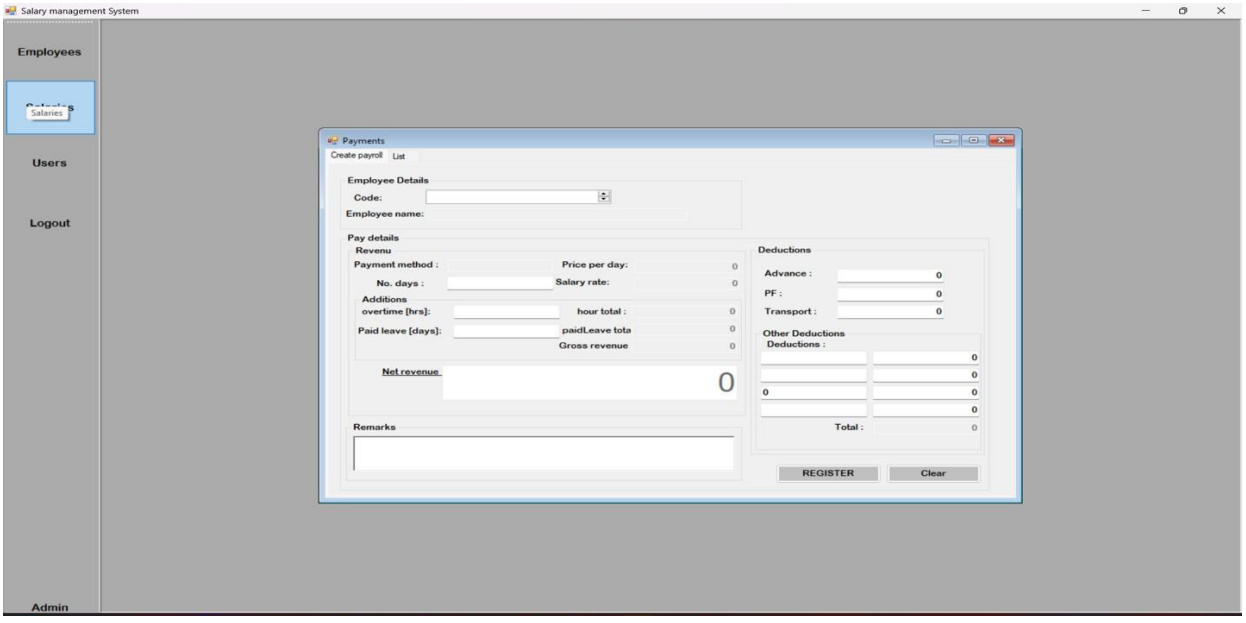
Employee Page:



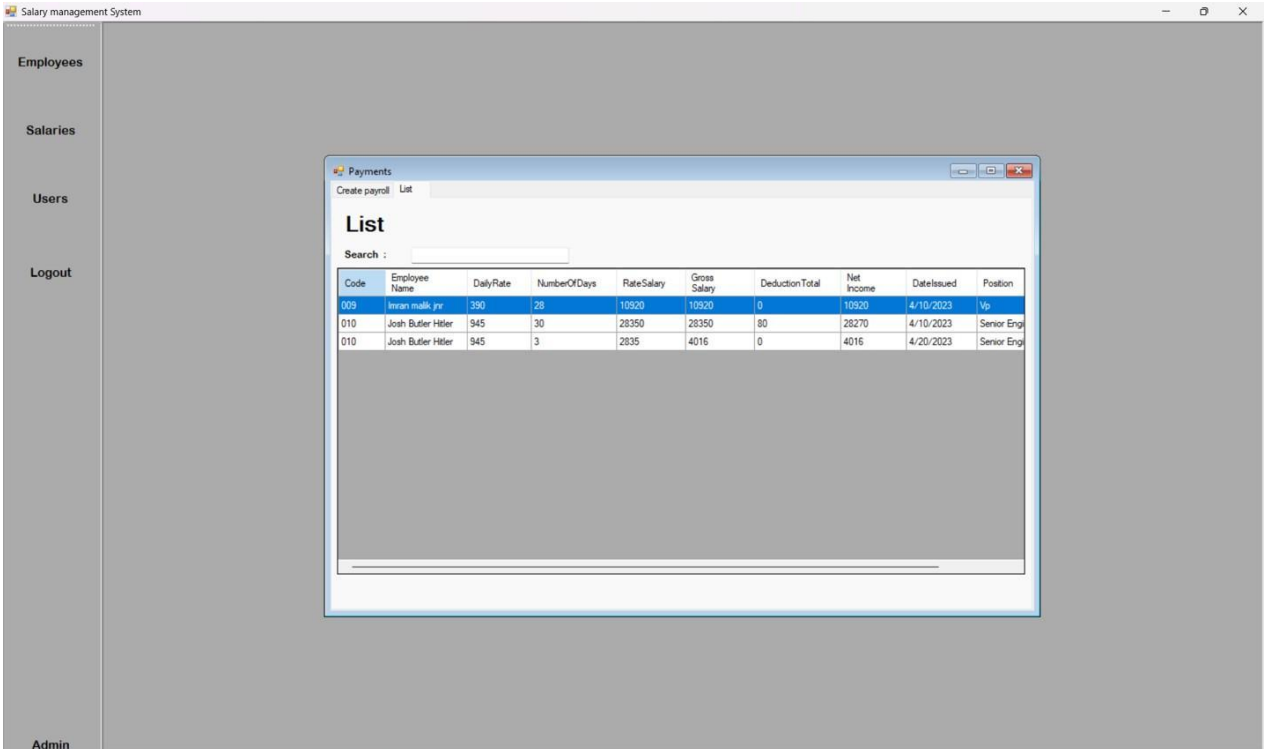
Employee list page;



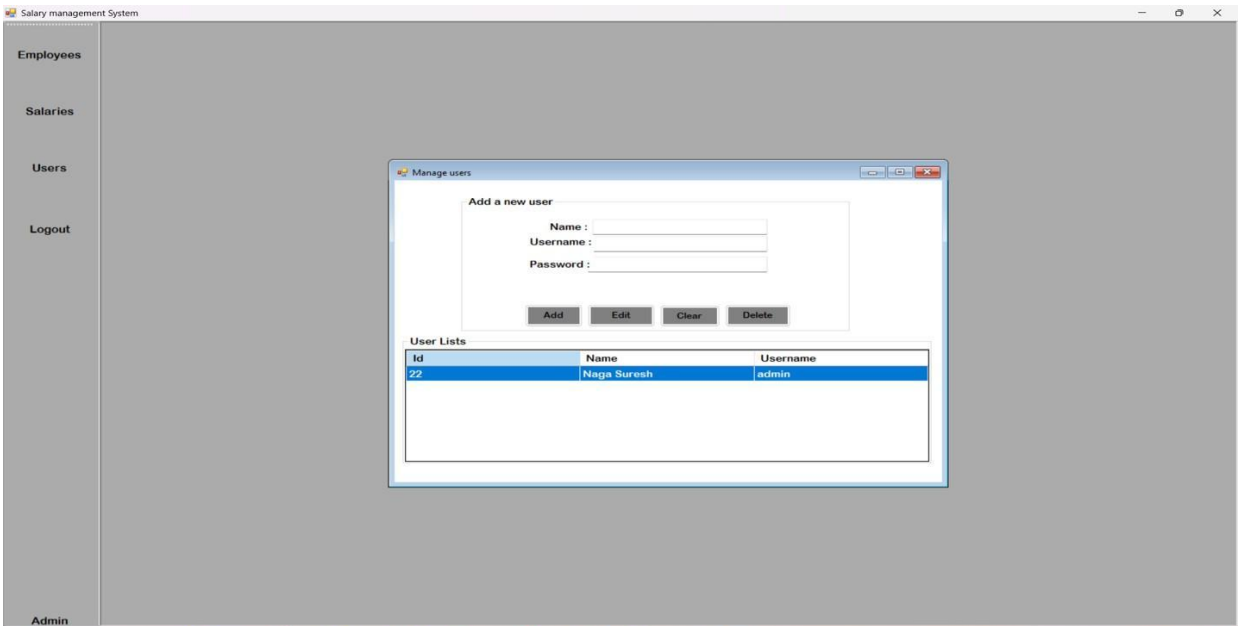
Salary Table page:



Salary list page:



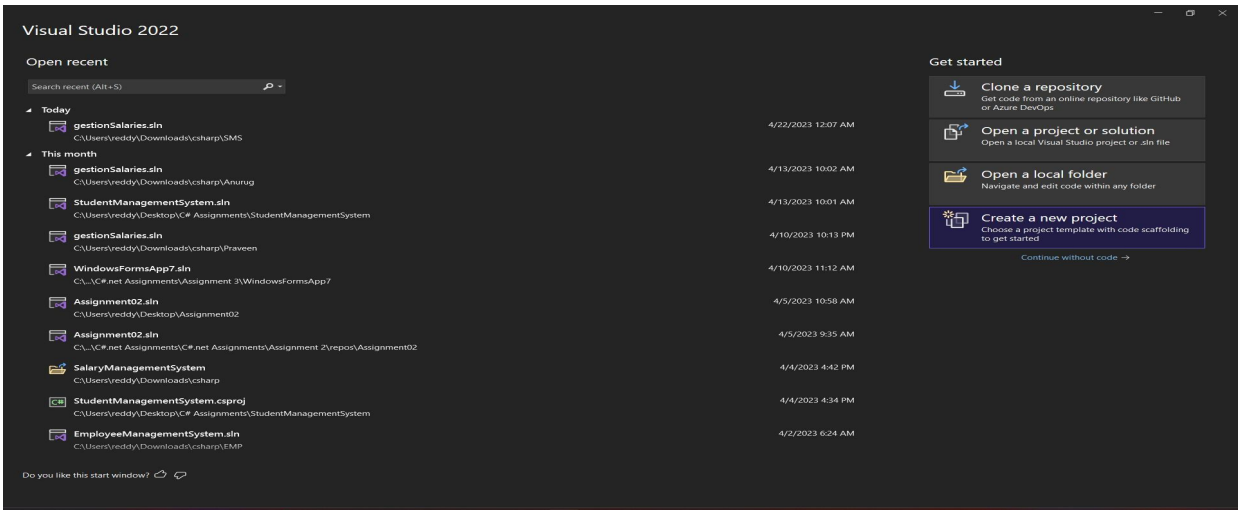
Users page:



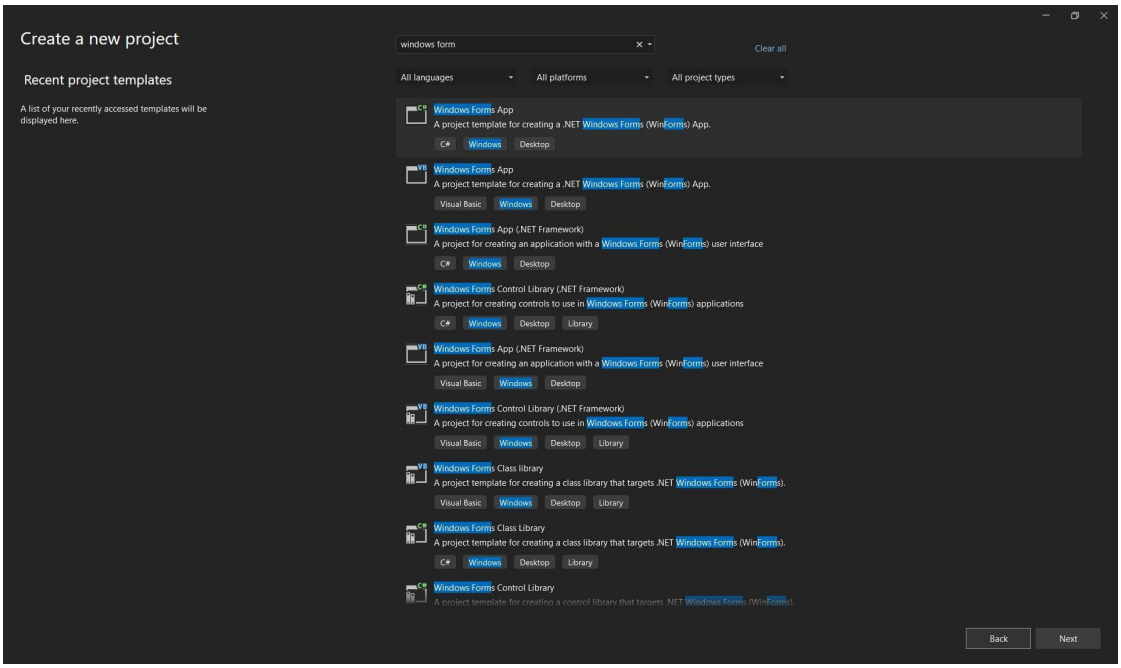
7. Steps for Development:

To develop an application we must install a visual studio. This may be any version like visual studio 2013 or higher.

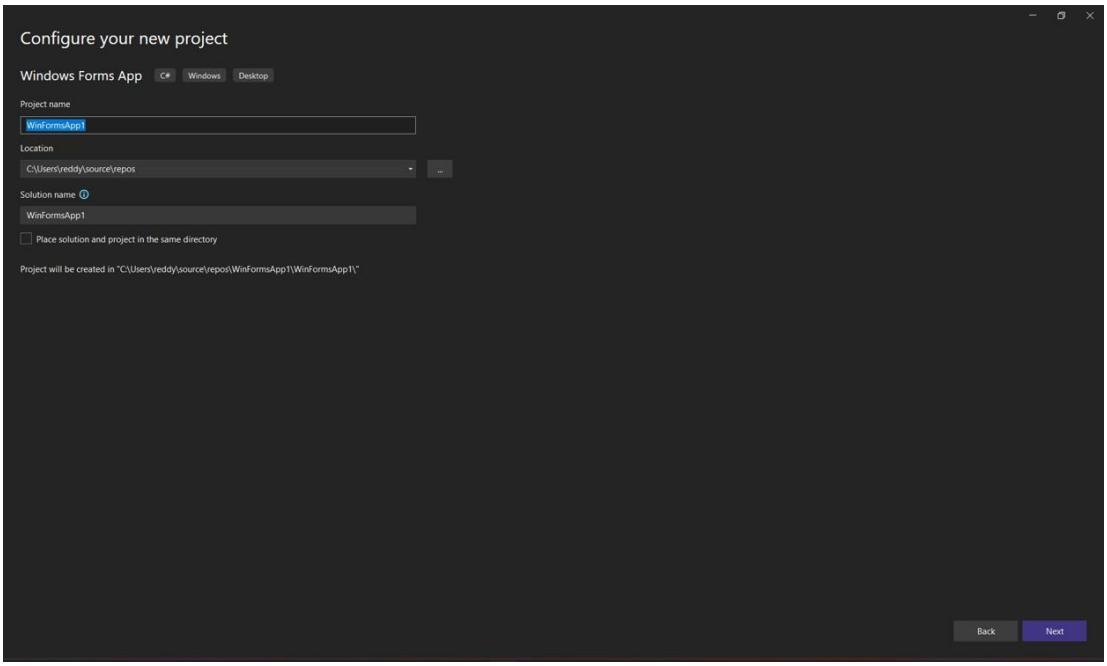
1. First, we have to go to **start -> visual studio 2017 -> File New project.**



2. Then, a menu will appear. Select visual c# web application.



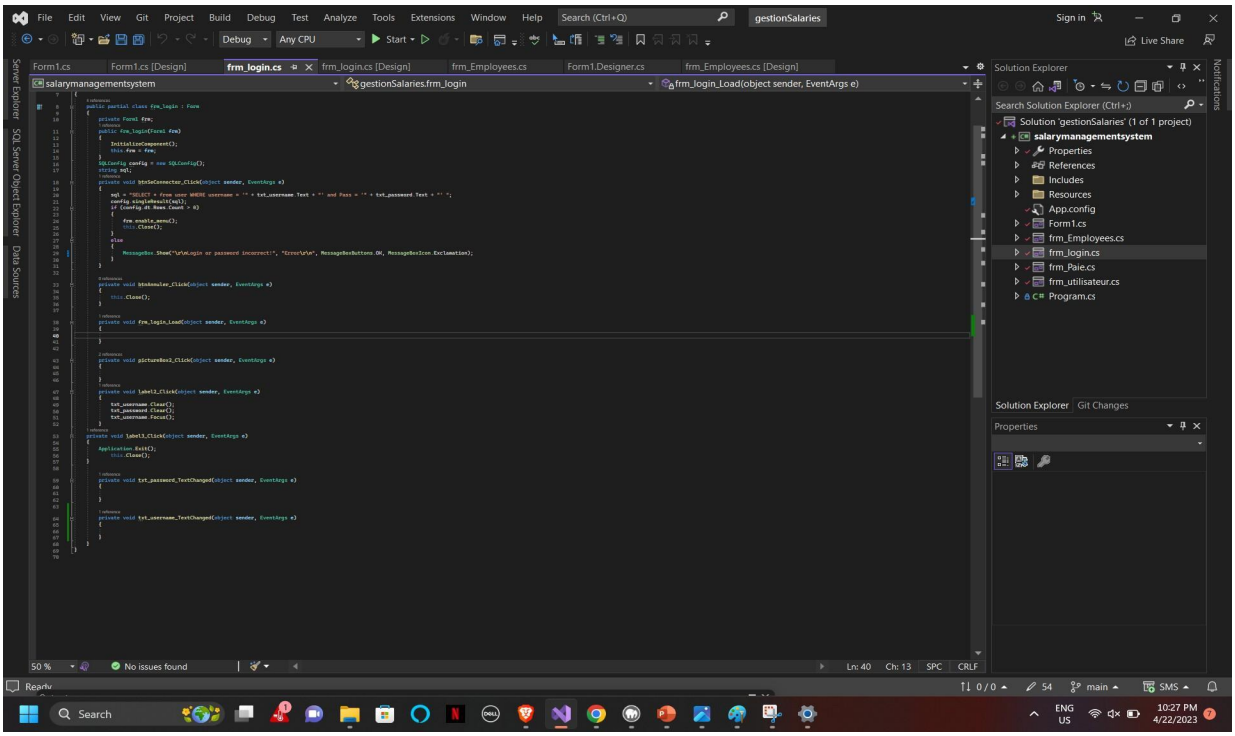
3. Again, click on web forms and click on ‘ok’.



4. Then write you code in that project.

8. Steps for deploying the project:

1. Login code



2. Code for employee

page using

gestionSalaries.Includes; using

System;

using System.Data;

using System.Windows.Forms;

namespace gestionSalaries

{

public partial class frm_Employees : Form

{

```
public frm_Employees()
{
   InitializeComponent();
}
SQLConfig config = new SQLConfig();
Fonctionutil funct = new Fonctionutil();
string sql;
string rdo;
private void btnempenregis_Click(object sender, EventArgs e)
{

   if (txtcode.Text == "" || txtfname.Text == "" || txtlname.Text == "" || txtmname.Text == ""
      || txtaddress.Text == "" || txtcontact.Text == "" || txtstatus.Text == "" || txtbplace.Text == ""
      || txtage.Text == "" || txtemerg.Text == "" || txtdrate.Text == "" || txtposition.Text == "")
   {
      MessageBox.Show("Please complete all fields !", "Info", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
   }
   else
   {
      if (rdomale.Checked)
      {
         rdo = "Male";
```

```
            }

            else

            {

                rdo = "Female";

            }

            sql = "INSERT INTO `employee_workinfo` (`emp_code`, `d_rate`, `p_method`,
`position`, `w_status`, `d_hired`)"

                    + " VALUES ('" + txtcode.Text + "','" + txtdrate.Text + "','" + txtpmethod.Text +
"','" + txtposition.Text

                    + "','" + txtworkstatus.Text + "','" + dtpdhired.Text + "')";

            config.Execute_Query(sql);



            sql = "INSERT INTO `employee` (`emp_code`, `emp_fname`, `emp_lname`,
`emp_mname`"

                + ", `address`, `contact`, `status`, `birth_date`, `birth_place`, `emp_sex`, `emp_age`"

                + ", `emerg_contct`) VALUES ('" + txtcode.Text + "','" + txtfname.Text + "','" +
txtlname.Text

                + "','" + txtmname.Text + "','" + txtaddress.Text + "'," + txtcontact.Text + ",'" + txtstatus.Text

                + "','" + dtpdbirth.Text + "','" + txtbplace.Text + "','" + rdo + "'," + txtage.Text + ",'" +
txtcontact.Text + "')";

            config.Execute_CUD(sql, "Error to execute.", "The data has been changed!");

            btnempnv_Click(sender, e);

        }



    }
```

```
private void txtcode_TextChanged(object sender, EventArgs e)
{
    sql = "SELECT * FROM `employee` e, `employee_workinfo` ew WHERE e.`emp_code`=ew.`emp_code` AND e.emp_code ='" + txtcode.Text + "'";
    config.singleResult(sql);
    if (config.dt.Rows.Count > 0)
    {
        txtdrate.Text = config.dt.Rows[0].Field<int>("d_rate").ToString();
        txtpmethod.Text = config.dt.Rows[0].Field<string>("p_method").ToString();
        txtposition.Text = config.dt.Rows[0].Field<string>("position").ToString();
        txtworkstatus.Text = config.dt.Rows[0].Field<string>("w_status").ToString();
        dtpdhired.Text = config.dt.Rows[0].Field<DateTime>("d_hired").ToString();

        txtfname.Text = config.dt.Rows[0].Field<string>("emp_fname").ToString();
        txtlname.Text = config.dt.Rows[0].Field<string>("emp_lname").ToString();
        txtmname.Text = config.dt.Rows[0].Field<string>("emp_mname").ToString();
        txtaddress.Text = config.dt.Rows[0].Field<string>("address").ToString();
        txtcontact.Text = config.dt.Rows[0].Field<long>("contact").ToString();
        txtstatus.Text = config.dt.Rows[0].Field<string>("status").ToString();
        dtpdbirth.Text = config.dt.Rows[0].Field<DateTime>("birth_date").ToString();
        txtbplace.Text = config.dt.Rows[0].Field<string>("birth_place").ToString();

        if (config.dt.Rows[0].Field<string>("emp_sex").ToString() == "Male")
```

```
    {
        rdomale.Checked = true;
    }
    else
    {
        rdofemale.Checked = true;

    }

    txtage.Text = config.dt.Rows[0].Field<int>("emp_age").ToString();
    txtcontact.Text = config.dt.Rows[0].Field<long>("emerg_contct").ToString();

    btnempenregis.Enabled = false;
    btnempupdate.Enabled = true;

}
else
{
    funct.clearTxt(GroupBox10);
    funct.clearTxt(GroupBox9);
    btnempenregis.Enabled = true;
    btnempupdate.Enabled =
    false;
}
```

```
}

private void btnempnv_Click(object sender, EventArgs e)
{
    funct.clearTxt(GroupBox10);
    funct.clearTxt(GroupBox9);

    btnempenregis.Enabled = true;
    btnempupdate.Enabled          =
    false; txtcode.Clear();

    list_Employee();
    funct.ResponsiveDtg(dtgemplist);
}

private void btnmodifier_Click(object sender, EventArgs e)

{
    // txtemerg.Text == ""
    if (txtcode.Text == "" || txtfname.Text == "" || txtlname.Text == "" || txtmname.Text == ""
      || txtaddress.Text == "" || txtstatus.Text == "" || txtbplace.Text == ""
      || txtage.Text == "" || txtcontact.Text == "" || txtdrate.Text == "" || txtposition.Text == "")
```

```
        {
            MessageBox.Show("\r\nPlease complete all fields !", "Info", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
        }
        else
        {
            if (rdomale.Checked)
            {
                rdo = "Male";
            }
            else
            {
                rdo = "Female";
            }

            sql = "UPDATE `employee_workinfo` SET `d_rate`='" + txtdrate.Text
              + "', `p_method`='" + txtpmethod.Text + "', `position`='" + txtposition.Text
              + "', `w_status`='" + txtworkstatus.Text + "', `d_hired`='" + dtpdhired.Text + "' WHERE
`emp_code`='" + txtcode.Text + "'";
            config.Execute_Query(sql);

            sql = "UPDATE `employee` SET `emp_fname`='" + txtfname.Text
            + "', `emp_lname`='" + txtlname.Text + "', `emp_mname`='" + txtmname.Text
            + "', `address`='" + txtaddress.Text + "', `contact`='" + txtcontact.Text + "', `status`='" +
txtstatus.Text
```

```
            + "', `birth_date`='" + dtpdbirth.Text + "', `birth_place`='" + txtbplace.Text + "',
`emp_sex`='" + rdo

            + "', `emp_age`='" + txtage.Text + "', `emerg_contct`='" + txtcontact.Text

            + "' WHERE `emp_code`='" + txtcode.Text + "'";

            config.Execute_CUD(sql, "Error to execute.", "The data has been changed!");




        }

    }



    private void list_Employee()

    {

        sql = "SELECT `emp_code` AS 'CODE',CONCAT( `emp_fname`,' ', `emp_lname`,' ',
`emp_mname`) AS 'NAME'"

            + ", `emp_age` AS 'AGE', `emp_sex` AS 'SEX', `status` AS 'STATUS', `address` AS
'ADDRESS'"

            + ", `contact` AS 'MOBILE'       FROM `employee` WHERE `emp_code` LIKE '%" +
txtempsearch.Text + "%'"

            + " OR emp_fname LIKE '%" + txtempsearch.Text + "%' OR emp_lname LIKE '%" +
txtempsearch.Text + "%'";

        config.Load_DTG(sql, dtgemplist);

    }



    private void frm_Employees_Load(object sender, EventArgs e)

    {

        btnempnv_Click(sender, e);
```

```
}

private void txtempsearch_TextChanged(object sender, EventArgs e)
{
    list_Employee();
}

private void btnsupp_Click(object sender, EventArgs e)
{
    sql = "DELETE FROM employee WHERE emp_code = '" + dtgemplist.CurrentRow.Cells[0].Value.ToString() + "'";
    config.Execute_Query(sql);

    sql = "DELETE FROM employee_workinfo WHERE emp_code = '" + dtgemplist.CurrentRow.Cells[0].Value.ToString() + "'";
    config.Execute_Query(sql);

    sql = "DELETE FROM other_deduction WHERE emp_code = '" + dtgemplist.CurrentRow.Cells[0].Value.ToString() + "'";
    config.Execute_Query(sql);

    MessageBox.Show("The employee has been successfully deleted!.\r\n", "Deleted", MessageBoxButtons.OK, MessageBoxIcon.Information);
    btnempnv_Click(sender, e);
}
```

```
private void btn_modifier_Click(object sender, EventArgs e)
{
    txtcode.Text = dtgemplist.CurrentRow.Cells[0].Value.ToString();
    TabControl2.SelectedTab = TabPage6;
    btnempenregis.Enabled = false;
    btnempupdate.Enabled = true;



}

private void Label26_Click(object sender, EventArgs e)
{


}

private void Label31_Click(object sender, EventArgs e)
{


}

private void Label20_Click(object sender, EventArgs e)
{
```

```
}

private void txtworkstatus_SelectedIndexChanged(object sender, EventArgs e)
{

}

private void TabPage6_Click(object sender, EventArgs e)
{

}

private void Label34_Click(object sender, EventArgs e)
{

}

private void GroupBox9_Enter(object sender, EventArgs e)
{

}
```

```
private void Label36_Click(object sender, EventArgs e)
{

}

private void Label21_Click(object sender, EventArgs e)
{

}

private void Label22_Click(object sender, EventArgs e)
{

}

private void GroupBox10_Enter(object sender, EventArgs e)
{

}

private void Label33_Click(object sender, EventArgs e)
{
```

```
        }

        private void txtemerg_TextChanged(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void txtcontact_TextChanged(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }
    }
}
```

3. Code for salary page.

```
using gestionSalaries.Includes; using System;
using System.Data;
using System.Windows.Forms;


namespace gestionSalaries
{
    public partial class frm_Paie : Form
    {
        public frm_Paie()
        {
            InitializeComponent();
        }


        SQLConfig config = new SQLConfig();
        Fonctionutil funct = new Fonctionutil();
        string sql;
```

```
private void calc_on_dedution()
{
    try
    {
        double ca, phic, pagibig, d1, d2, d3, d4, total_deduction, gross, total_net;

        ca = double.Parse(txtpcadvance.Text);
        phic = double.Parse(txtpphic.Text);
        pagibig = double.Parse(txtppagibig.Text);
        d1 = double.Parse(txtpdeduct1.Text);
        d2 = double.Parse(txtpdeduct2.Text);
        d3 = double.Parse(txtpdeduct3.Text);
        d4 = double.Parse(txtpdeduct4.Text);
        gross = double.Parse(txtpgincome.Text);

        total_deduction = ca + phic + pagibig + d1 + d2 + d3 + d4;
        txtpdeducttot.Text = total_deduction.ToString();

        total_net = gross - total_deduction;
        txtpnetincome.Text = total_net.ToString();
    }
    catch
```

```
        {


        }


    }


    private void load_data()

    {

        sql = "SELECT DISTINCT (" +

         "p.`trans_id`" +

         "),e.emp_code as 'Code', CONCAT( `emp_fname` , ' ', `emp_lname` , ' ', `emp_mname`
) AS 'Employee Name'" +

         ", `d_rate` AS 'DailyRate' , `num_days` AS 'NumberOfDays' , `r_wage` AS 'RateSalary',
"

         + " `gross_sal` AS 'Gross Salary', `total_ded` AS 'DeductionTotal', `net_income` AS 'Net
Income' , " +

         " `dateissued` AS 'DateIssued' , `position` AS 'Position' ,`remarks` AS 'Remarks' " +

         "FROM `employee` e, `payroll` p, `employee_workinfo` ew, `other_deduction` od " +

         "WHERE e.`emp_code` = p.`emp_code` " +

         "AND p.`emp_code` = ew.`emp_code` " +

         "AND p.`trans_id` = od.`trans_id` " +

         "AND (emp_fname LIKE '%" + txtpsearch.Text + "%'" +

        "OR emp_lname LIKE '%" + txtpsearch.Text + "%'" + "OR

        e.emp_code LIKE '%" + txtpsearch.Text + "%')";

        config.Load_DTG(sql, dtgParollList);
```

```
    dtgParollList.Columns[0].Visible = false;
    funct.ResponsiveDtg(dtgParollList);



    sql = "SELECT concat(autoname,strnum) as auto FROM autonumber WHERE id = 1";
    config.singleResult(sql);
    if (config.dt.Rows.Count > 0)
    {
        txttrancode.Text = config.dt.Rows[0].Field<string>("auto");
    }
}

private void Button2_Click(object sender, EventArgs e)
{
    txtPEmployeeName.Text = "";
    txtpremarks.Text = "";
    txtPAssignCode.Text = "";
    funct.clearTxt(GroupBox3);
    funct.clearTxt(GroupBox4);
    funct.clearTxt(GroupBox5);
    funct.clearTxt(GroupBox6);
    txtPrateWage.Text = "0";
    txtPregOt.Text = "0";
```

```
    txtPholPay.Text = "0";
    txtpgincome.Text = "0";
    txtpnetincome.Text = "0";
    txtpdeducttot.Text = "0";
    txtpcadvance.Text = "0";
    txtpphic.Text = "0";
    txtppagibig.Text = "0";
    txtpdeduct1.Text = "0";
    txtpdeduct2.Text = "0";
    txtpdeduct3.Text = "0";
    txtpdeduct4.Text = "0";



}

private void frm_Paie_Load(object sender, EventArgs e)
{
    load_data();




}
```

```
private void txtPAssignCode_TextChanged(object sender, EventArgs e)
{

    sql = "SELECT * FROM `employee` e, `employee_workinfo` ew "
      + " WHERE e.`emp_code`=ew.`emp_code` AND e.emp_code ='" + txtPAssignCode.Text + "'";
    config.singleResult(sql);
    if (config.dt.Rows.Count > 0)
    {
      txtPRateperday.Text = config.dt.Rows[0].Field<int>("d_rate").ToString();
      txtPPayPeriod.Text = config.dt.Rows[0].Field<string>("p_method").ToString();
      txtPEmployeeName.Text = config.dt.Rows[0].Field<string>("emp_fname").ToString()
      + " " + config.dt.Rows[0].Field<string>("emp_lname").ToString() + " "
      + config.dt.Rows[0].Field<string>("emp_mname").ToString();
    }
    else
    {
      txtPEmployeeName.Text = "";
      txtpremarks.Text = "";
      funct.clearTxt(GroupBox3);
      funct.clearTxt(GroupBox4);
      funct.clearTxt(GroupBox5);
      funct.clearTxt(GroupBox6);
      txtPrateWage.Text = "0";
```

```
        txtPregOt.Text = "0";
        txtPholPay.Text = "0";
        txtpgincome.Text = "0";
        txtpnetincome.Text = "0";
        txtpdeducttot.Text = "0";
    }

}

private void txtPNoDays_TextChanged(object sender, EventArgs e)
{
    try
    {

        double rateWage, grossincome, neticome, ot, holpay;

        ot = double.Parse(txtPregOt.Text);
        holpay = double.Parse(txtPholPay.Text);

        if (txtPNoDays.Text == "" || txtPNoDays.Text == "0")
        {
            txtPrateWage.Text = "0";
            rateWage = double.Parse(txtPrateWage.Text);
```

```
    }
    else
    {
        rateWage = double.Parse(txtPRateperday.Text) * double.Parse(txtPNoDays.Text);
        txtPrateWage.Text = rateWage.ToString();

    }




    grossincome = rateWage + ot + holpay;
    txtpgincome.Text = grossincome.ToString();



    neticome = grossincome - double.Parse(txtpdeducttot.Text);
    txtpnetincome.Text = neticome.ToString();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

```
}

private void txtPRegOtHr_TextChanged(object sender, EventArgs e)
{
   try
   {



      double total, total_OT, grossincome, neticome, ot, holpay, rateWage;



      if (txtPRegOtHr.Text == "" || txtPRegOtHr.Text == "0")
      {
         txtPregOt.Text = "0";
      }
      else
      {
         total = Double.Parse(txtPRateperday.Text) / 8;
         total_OT = total * Double.Parse(txtPRegOtHr.Text);
         txtPregOt.Text = total_OT.ToString();


      }
```

```
        ot = double.Parse(txtPregOt.Text);
        holpay = double.Parse(txtPholPay.Text);
        rateWage = double.Parse(txtPrateWage.Text);

        grossincome = rateWage + ot + holpay;
        txtpgincome.Text = grossincome.ToString();

        neticome = grossincome - double.Parse(txtpdeducttot.Text);
        txtpnetincome.Text = neticome.ToString();

    }
    catch
    {

    }
}

private void txtPholPayDay_TextChanged(object sender, EventArgs e)
{
    try
    {
```

```
double total_hol, grossincome, neticome, ot, holpay, rateWage;

if (txtPholPayDay.Text == "" || txtPholPayDay.Text == "0")
{
   txtPholPay.Text = "0";
}
else
{
   total_hol = double.Parse(txtPRateperday.Text) * double.Parse(txtPholPayDay.Text);

   txtPholPay.Text = total_hol.ToString();

}

ot = double.Parse(txtPregOt.Text);
holpay = double.Parse(txtPholPay.Text);
rateWage = double.Parse(txtPrateWage.Text);

grossincome = rateWage + ot + holpay;
txtpgincome.Text = grossincome.ToString();
```

```
        neticome = grossincome - double.Parse(txtpdeducttot.Text);
        txtpnetincome.Text = neticome.ToString();

    }
    catch
    {

    }

}

private void txtpcadvance_TextChanged(object sender, EventArgs e)
{


    if (txtpcadvance.Text == "" || txtpcadvance.Text == "0")
    {
        txtpcadvance.Text = "0";

    }

    calc_on_dedution();
```

```
}

private void txtpphic_TextChanged(object sender, EventArgs e)
{
    if (txtpphic.Text == "" || txtpphic.Text == "0")
    {
        txtpphic.Text = "0";

    }
    calc_on_dedution();
}

private void txtppagibig_TextChanged(object sender, EventArgs e)

{
    if (txtppagibig.Text == "" || txtppagibig.Text == "0")
    {
        txtppagibig.Text = "0";

    }
    calc_on_dedution();
}
```

```
private void txtpdeduct1_TextChanged(object sender, EventArgs e)
{
    if (txtpdeduct1.Text == "" || txtpdeduct1.Text == "0")
    {
        txtpdeduct1.Text = "0";

    }
    calc_on_dedution();
}

private void txtpdeduct2_TextChanged(object sender, EventArgs e)
{
    if (txtpdeduct2.Text == "" || txtpdeduct2.Text == "0")
    {
        txtpdeduct2.Text = "0";

    }
    calc_on_dedution();
}

private void txtpdeduct3_TextChanged(object sender, EventArgs e)
{
```

```
    if (txtpdeduct3.Text == "" || txtpdeduct3.Text == "0")
    {
        txtpdeduct3.Text = "0";

    }
    calc_on_dedution();

}

private void txtpdeduct4_TextChanged(object sender, EventArgs e)
{
    if (txtpdeduct4.Text == "" || txtpdeduct4.Text == "0")
    {
        txtpdeduct4.Text = "0";

    }
    calc_on_dedution();
}

private void btnEnregistrer_Click(object sender, EventArgs e)
{

    if (txtPNoDays.Text == "" || txtPregOt.Text == "" || txtPholPay.Text == "")
```

```
        {
            MessageBox.Show("Please complete all fields !", "Info", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
        }
        else
        {
            sql = "INSERT INTO `other_deduction` "
                    +  "(`emp_code`,  `deduct1`,  `ded_amount1`,  `deduct2`,  `ded_amount2`,
`deduct3`, "
                    + "`ded_amount3`, `deduct4`, `ded_amount4`, `total_ded`,trans_id)"
                    + " VALUES ('" + txtPAssignCode.Text + "','" + txtpdeductname1.Text + "','"
                    + txtpdeduct1.Text + "','" + txtpdeductname2.Text + "','" + txtpdeduct2.Text
                    + "','" + txtpdeductname3.Text + "','" + txtpdeductname3.Text
                    + "','" + txtpdeductname4.Text + "','" + txtpdeductname4.Text
                    + "','" + txtpdeducttot.Text + "','" + txttrancode.Text + "')";
            config.Execute_Query(sql);

            sql = "INSERT INTO `payroll` "
               + "(`emp_code`, `num_days`, `r_wage`, `overtime`, `hol_pay`, `gross_sal`"
               +  ",  `cash_ad`,  `bread_vale`,  `philhealth`,  `pag-ibig`,  `net_income`,
`remarks`,`dateissued`,trans_id)"
               + " VALUES ('" + txtPAssignCode.Text + "','" + txtPNoDays.Text + "','"
               + txtPrateWage.Text + "','" + txtPregOt.Text + "','" + txtPholPay.Text
               + "','" + txtpgincome.Text + "','" + txtpcadvance.Text + "','" + txtbvale.Text
               + "','" + txtpphic.Text + "','" + txtppagibig.Text + "','" + txtpnetincome.Text
```

```
            + "','" + txtpremarks.Text + "',Now(),'" + txttrancode.Text + "')";


            config.Execute_Query(sql);


            MessageBox.Show("The data has been changed!");


            config.Execute_Query("UPDATE autonumber SET strnum = strnum + increment
WHERE id =1");
            txtPAssignCode.Text = "";


            load_data();




        }


    }


    private void txtpsearch_TextChanged(object sender, EventArgs e)
    {
        load_data();
    }


    private void Label6_Click(object sender, EventArgs e)
    {
```

```
}

private void Label9_Click(object sender, EventArgs e)
{

}

private void Label10_Click(object sender, EventArgs e)
{

}

private void GroupBox3_Enter(object sender, EventArgs e)
{

}

private void Label11_Click(object sender, EventArgs e)
{

}
```

```
private void Label1_Click(object sender, EventArgs e)
{

}

private void Label5_Click(object sender, EventArgs e)
{

}

private void GroupBox6_Enter(object sender, EventArgs e)
{

}

private void txtPregOt_TextChanged(object sender, EventArgs e)
{

}

private void Label4_Click(object sender, EventArgs e)
{
```

```
}

private void txtpdeducttot_TextChanged(object sender, EventArgs e)
{

}

private void GroupBox4_Enter(object sender, EventArgs e)
{

}

private void Label8_Click(object sender, EventArgs e)
{

}

private void Label19_Click(object sender, EventArgs e)
{

}

private void TabPage8_Click(object sender, EventArgs e)
```

```
        {


        }


        private void Label15_Click(object sender, EventArgs e)
        {


        }


        private void GroupBox5_Enter(object sender, EventArgs e)
        {


        }
    }
}
```

9. References:

1. "Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design" by Michael J. Hernandez. This book provides a practical guide to database design and is a great resource for beginners.

2. "SQL for Data Analysis: Beginner's Guide for Business Intelligence, Data Science & Data Analytics" by David Feldspar. This book introduces SQL and covers basic and advanced SQL topics.

3. "SQL Cookbook: Query Solutions and Techniques for All SQL Users" by Anthony Molinaro. This book provides a collection of SQL recipes for solving common problems.

4. "Microsoft SQL Server Documentation" (https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15). This documentation provides information on SQL Server and includes tutorials, reference materials, and troubleshooting guides.

5. "Oracle Database Documentation" (https://docs.oracle.com/en/database/). This documentation provides information on Oracle Database and includes tutorials, reference materials, and troubleshooting guides.

6. "MySQL Documentation" (https://dev.mysql.com/doc/). This documentation provides information on MySQL and includes tutorials, reference materials, and troubleshooting guides.