MOTILAL DULICHAND PVT. LTD.

A Project Report

Submitted As Part of the Internship Program

Project Titled:-Payroll System

Submitted By

Vanshika Thavnani

Under The Guidance Of

Mr.Ram Chander

20, Industrial Estate, Kalpi Road,

Shastri Nagar, Kanpur Nagar-208012,

Uttar Pradesh, India

ACKNOWLEGEMENT

I am immensely grateful to Motilal Dulichand for providing me with the opportunity to undertake my internship and complete the project titled **"Payroll Management System"** during the period from 3 June 2024 to 17 July I extend my heartfelt thanks to **Mr. Ram Chander, Head of EDP Department,** whose unwavering guidance, expertise, and encouragement were invaluable throughout the project. Their insightful feedback and constructive criticism greatly contributed to refining the project and enhancing my skills in software development and payroll management.

I would also like to acknowledge the support and cooperation received from the entire team at Motilal Dulichand. Their willingness to share knowledge, provide assistance, and offer practical solutions were instrumental in overcoming challenges and achieving project milestones effectively.

Lastly, I express my appreciation to all those who supported and contributed to my internship experience at Motilal Dulichand.

NAME: Vanshika Thavnani
COLLEGE NAME: Kristu Jayanti College,Autonomous Bengaluru
DATE: 17 July 2024

ABSTRACT

This documentation provides a comprehensive overview of a payroll system developed during an internship at Motilal Dulichand, utilizing Visual Basic (VB) for frontend development and SQL for backend database management. The project aimed to enhance efficiency in payroll operations by automating employee data management, salary calculations based on predefined structures, and generating detailed reports for department-wise salary distributions. The system architecture was designed to ensure scalability and reliability, with VB offering an intuitive user interface for HR administrators to seamlessly manage employee records, while SQL facilitated secure storage and efficient retrieval of data. Key components covered include a detailed exploration of the database schema, outlining tables, relationships, and data structures employed to manage employee information and salary computations. This project not only enriched technical skills in software development and database integration but also highlighted the practical application of technology in optimizing organizational processes and decision-making. Overall, the payroll system project exemplifies the successful application of theoretical knowledge in a real-world setting, underscoring its significance in professional growth and proficiency in information technology.

TABLE OF CONTENTS

CONTENT OF TABLES

INTRODUCTION

During my internship at Motilal Dulichand, I undertook the development of a comprehensive payroll management system using Visual Basic (VB) and SQL. This project served as a practical learning opportunity to deepen my understanding of software development and database integration. The system was designed to automate the management of employee details, salary structures, and reporting processes. Despite being a training project, its objectives included creating a user-friendly interface for HR administrators to efficiently handle employee records, automate salary calculations based on predefined rules, and generate insightful reports.

While the system was not implemented for operational use within the company, its development provided valuable insights into the complexities of payroll management and software engineering practices. By leveraging VB for frontend development, I focused on creating an intuitive interface that ensured seamless interaction with the system. On the backend, SQL facilitated robust data management, ensuring secure storage and efficient retrieval of employee and salary information. This technology stack not only met the project's technical requirements but also equipped me with practical skills in application development and database administration.

Throughout the project, I collaborated closely with HR professionals to align the system's functionalities with industry practices and organizational needs. This experience not only enhanced my technical proficiency but also underscored the importance of user feedback in refining software solutions. By simulating real-world payroll scenarios and addressing specific challenges encountered during development, I gained valuable hands-on experience in software lifecycle management and project execution within a corporate environment.

In conclusion, while the payroll management system developed during my internship at Motilal Dulichand was primarily for training purposes, it provided a solid foundation in software development and database integration. This project not only sharpened my technical skills but also deepened my appreciation for the role of technology in optimizing business processes. It remains a significant milestone in my journey toward becoming a proficient software developer and problem solver in the field of information technology.

DATABASE SCHEMA

A database schema is like a blueprint that shows how data is organized in a database. It includes tables with information, how these tables are connected, and rules to keep the data accurate. The database schema for the payroll system I developed during my internship at Motilal Dulichand is very important for making sure everything works well and efficiently. This schema is created using SQL and is designed to store and manage key employee information and salary data, working seamlessly with the Visual Basic frontend.

Here are the main parts of the schema:

- Employee Table: This table stores personal and job-related details about employees. It helps in easily accessing and managing employee information.
- Salary Table: This table contains details about each employee's salary, including different components like basic pay and allowances. It ensures that salaries are calculated correctly and fairly.
- Transaction Table: This table records all salary-related transactions, such as total salary, loans, advances, and deductions. It helps in keeping track of all the financial transactions related to payroll.
- Parameter Table: This table includes various payroll parameters, like rates and limits for benefits and deductions. It helps in maintaining and updating payroll rules and rates according to company policies and legal requirements.

These tables are linked together using keys. For example, the Employee table is connected to the Department table, making sure that each employee is linked to a valid department. Similarly, the Salary table is linked to the Employee table to connect salary details with the right employee. These links help keep the data accurate and make it easier to retrieve complex information.

The schema also has rules to ensure data accuracy. For instance, each employee must have a unique identifier, and salary values should be valid. These rules help prevent errors and keep the data consistent.

This well-designed database schema makes sure that data is stored and retrieved securely and efficiently, following industry standards. It highlights the technical details and careful

planning needed to meet the organization's needs. The schema ensures the payroll system can handle different scenarios, adapt to changes, and support future improvements, providing a reliable solution for managing employee payroll.

Here is the table structure of the tables used:

1-EMPLOYEE

FIELD	TYPE	NULL	KEY
Employee Code	Numeric(18, 0)	Not Null	Primary Key
Employee Name	Varchar(50)	Null	
Gender	Varchar(50)	Null	
Address	Varchar(50)	Null	
Dob	Date	Null	
City	Varchar(50)	Null	
Qualification	Varchar(50)	Null	
DOJ	Date	Null	
Shift	Time(7)	Null	
Pin code	Numeric(18, 0)	Null	
Phone	Numeric(18, 0)	Null	
Email	Varchar(50)	Null	
Pf_Number	Numeric(18, 0)	Null	
Esi_Number	Numeric(18, 0)	Null	
Department	Varchar(50)	Null	

TABLE 1:EMPLOYEE

2-SALARY

FIELD	TYPE	NULL	KEY
Employee Code	Numeric(18, 0)	Not Null	Foreign Key
Department	Varchar(50)	Null	
Designation	Varchar(50)	Null	
Basic_Salary	Decimal(18, 0)	Null	
Da	Decimal(18, 0)	Null	
Hra	Decimal(18, 0)	Null	
Medical	Decimal(18, 0)	Null	
Performance	Decimal(18, 0)	Null	
Conveyance	Decimal(18, 0)	Null	

TABLE 2:SALARY

3-TRANSACTION

FIELD	TYPE	NULL	KEY
Employee Code	Numeric(18, 0)	Not Null	Primary Key
Month	Varchar(50)	Null	
Year	Int	Null	
Working_Day	Int	Null	
Holiday	Int	Null	
Early_Leave	Int	Null	
Casual_Leave	Int	Null	
Leave_Without_Pay	Int	Null	
Tds	Int	Null	
Advance	Int	Null	
Loan_From_Bank	Decimal(18, 0)	Null	
Loan_From_Company	Decimal(18, 0)	Null	
Total_Salary	Decimal(18, 0)	Null	

TABLE 3:TRANSACTION

4-PARAMETER

FIELD	TYPE	NULL	KEY
Pf Limit	Decimal(18, 0)	Null	
Esi Limit	Decimal(18, 0)	Null	
Pf Employee Rate	Decimal(18, 0)	Null	
Pf Employer Rate	Decimal(18, 0)	Null	
Esi Employee Rate	Decimal(18, 0)	Null	
Esi Employer Rate	Decimal(18, 0)	Null	
Pf Type	Varchar(50)	Null	
Esi Type	Varchar(50)	Null	

TABLE 4:TRANSACTION

5-COMPANY

FIELD	TYPE	NULL	KEY
Company Name	Varchar(50)	Null	
Address	Varchar(Max)	Null	
Phone	Numeric(18, 0)	Null	
Email	Varchar(50)	Null	

TABLE 5:COMPANY

ER DIAGRAM

An Entity-Relationship (ER) diagram is a visual representation of the data and the relationships between data within a database. It is a crucial tool in database design and helps in understanding how different entities (objects) interact with each other within the system.

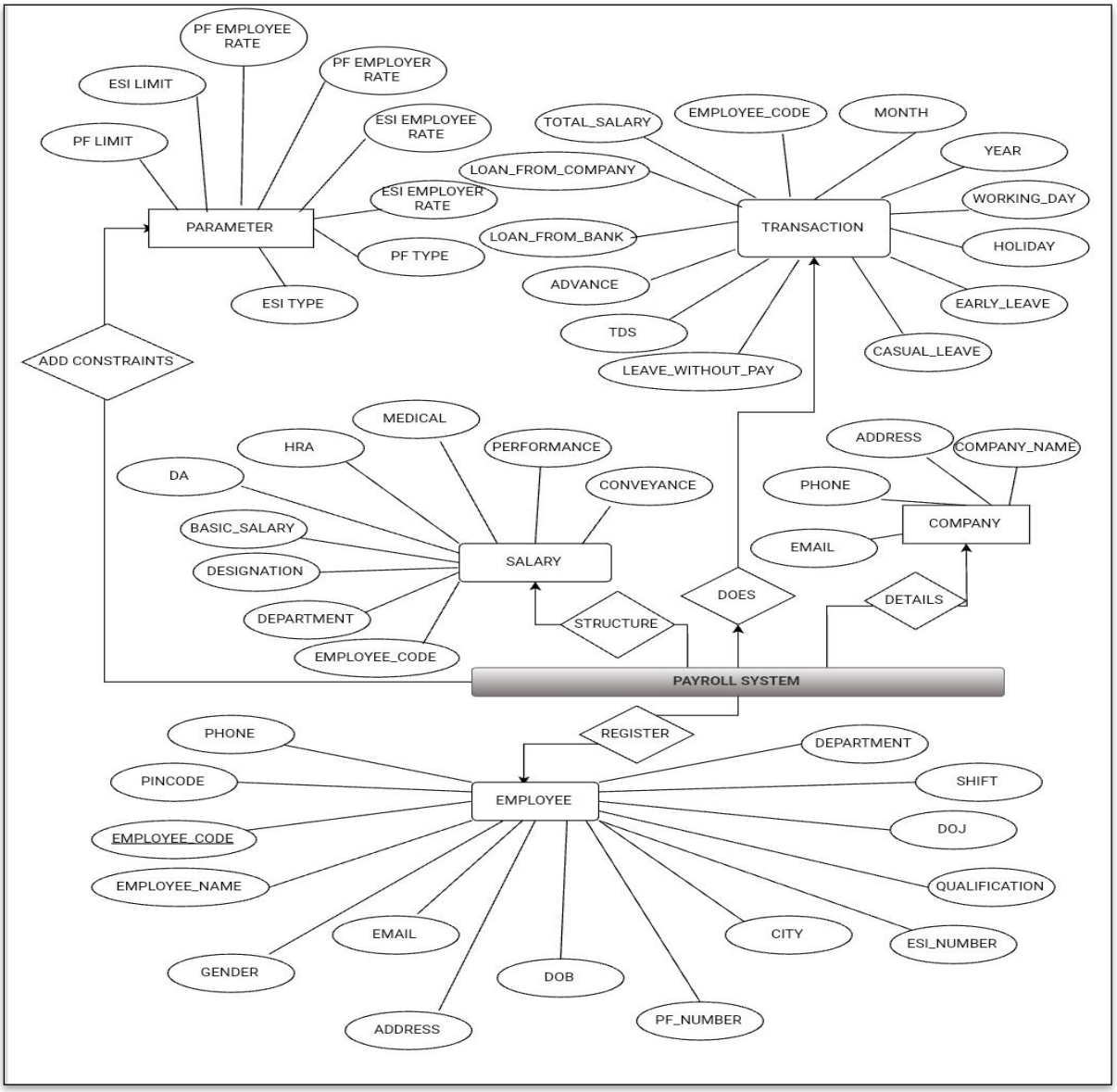
COMPONENTS AND SYMBOLS OF AN ER DIAGRAM

1. **Entity**
    - **Symbol**: Rectangles
    - **Description**: Entities represent real-world objects or concepts that have data stored about them in the database. Examples include "Employee," and "Salary."
2. **Attributes**
    - **Symbol**: Ovals
    - **Description**: Attributes are the properties or details of an entity. For instance, an "Employee" entity may have attributes such as Name, Address, and Phone Number.
3. **Relationships**
    - **Symbol**: Diamonds
    - **Description**: Relationships show how entities are related to each other. The relationship is usually labeled with verbs like "works in," "has," "belongs to," etc.
4. **Primary Key**
    - **Symbol**: Underlined text inside the entity rectangle
    - **Description**: A primary key is a unique identifier for an entity. It ensures that each record within the entity is unique. For example, EmployeeID can be the primary key for the Employee entity.
5. **Cardinality**
    - **Symbol**: Notations like 1:1, 1, N
    - **Description**: Cardinality defines the numerical relationship between entities:
        - **1:1 (One-to-One)**: One entity instance is associated with one instance of another entity.

- **1(One-to-Many)**: One entity instance is associated with multiple instances of another entity.
- **N(Many-to-Many)**: Multiple instances of one entity are associated with multiple instances of another entity.

Understanding ER diagrams and their symbols is essential for designing efficient and well-structured databases, ensuring data integrity, and supporting the system's operational needs effectively.

ER DIAGRAM FOR PAYROLL SYSTEM



USER INTERFACE

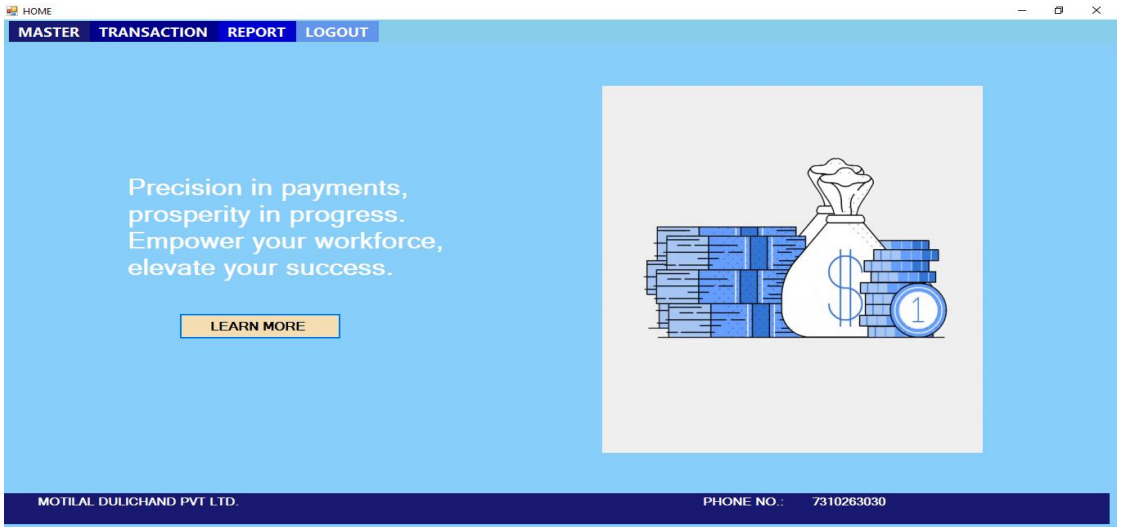
User Interface (UI) in the context of software refers to the visual elements, controls, and interactions through which a user interacts with a system. It encompasses the layout, design, and functionality of the graphical interface that users interact with to perform tasks and access information within an application or system.

Displayed below are the current screenshots of the payroll system's forms, provided for reference and evaluation.

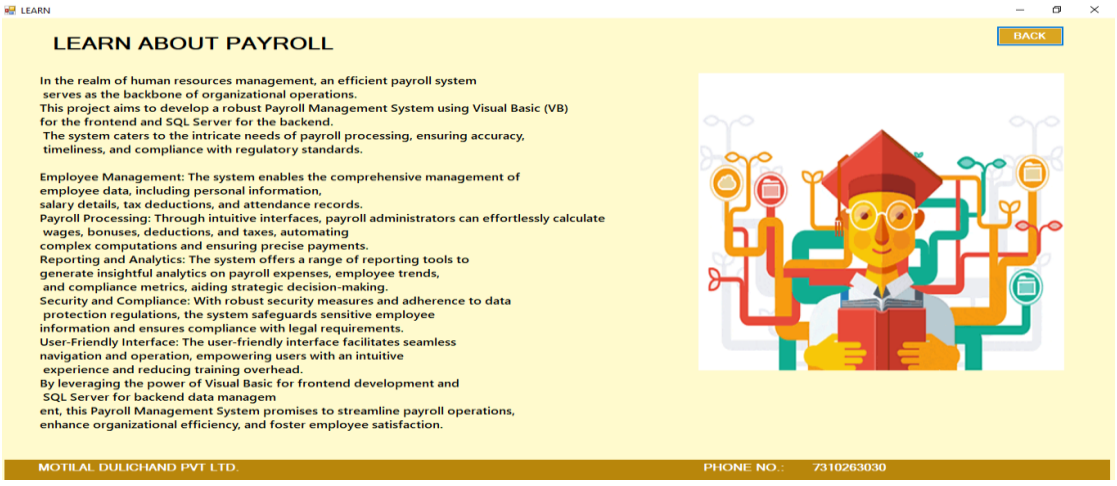
FORM1:-

LOGIN:-

HOME PAGE:-



LEARN:-



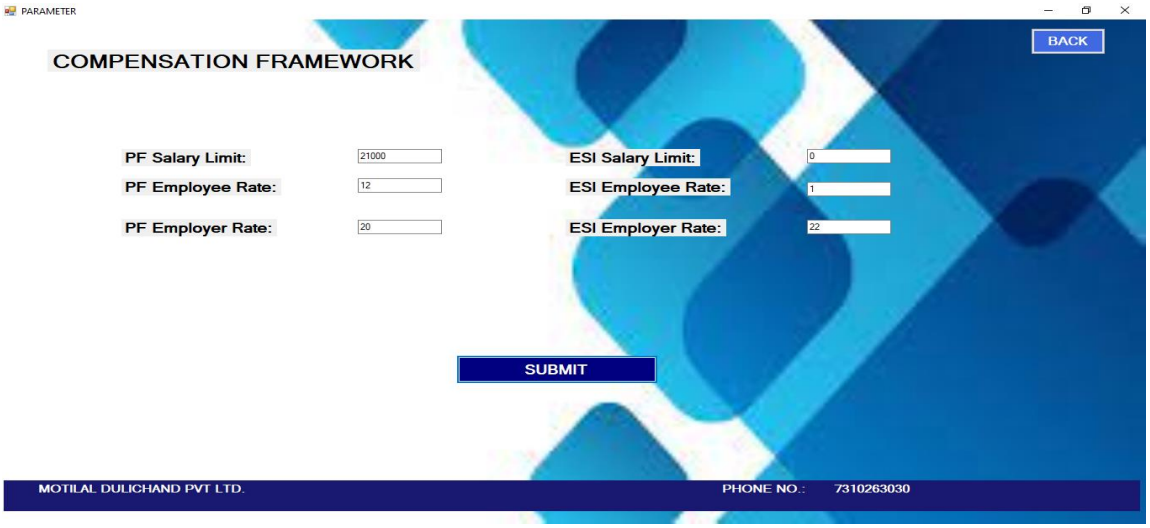
ADD EMPLOYEE DETAILS:-

EMPLOYEE DETAILS:-

EMPLOYEE_C	EMPLOYEE_N	GENDER	ADDRESS	DOB	CITY	QUALIFICATIC	DOJ	SHIFT	PINCODE	PHONE	EMAIL	PF_NUMBER	ESI_NUMBER	DEPARTMEN
11133	RAHUL GU...	MALE	56 SP LANE	22/03/1990	NEW DELHI	B.TECH	30/05/2024	16:00:00	1100001	9812345678	RAHUL@G...	1003	2003	INFORMATI...
17265	PRIYA PATEL	FEMALE	34 SV STRE...	22/07/1989	AHMEDABAD	M.A.	15/03/2023	12:00:00	380001	9876543210	PRIYA@GM...	1002	2002	HUMAN RE...
27747	SUMAN	FEMALE	GUMTI NO.5	16/08/2000	KANPUR	BBA	07/06/2024	12:00:00	208009	45367922	SUMAN@G...	34256	7765	HUMAN RE...
27770	VIVEK NAIR	MALE	22 KL LANE	07/03/1990	KOCHI	B.E.	14/06/2024	09:00:00	682001	9847034567	VIVEK@GM...	1007	2007	INFORMATI...
30223	POOJA IYER	FEMALE	33 TN STR...	12/03/1999	CHENNAI	M.B.A	30/05/2024	12:00:00	600001	9784234561	POJA@GM...	1008	2008	PRODUCTI...
34002	ANJALI VER...	FEMALE	78 RK ROAD	11/02/2000	BANGALORE	B..A.	13/07/2023	09:00:00	560001	9845123456	ANJALI@G...	1004	2004	SALES
36361	VANSH	MALE	SHASTRI N...	16/06/2001	KANPUR	B.COM	03/07/2024	12:00:00	208005	9555666412	VANSH@G...	123456	789456	FINANCE
40356	SUNITA JO...	FEMALE	55 PB LANE	13/03/2009	CHANDIGA...	B.A.	06/06/2024	09:00:00	1600001	965467894	SUNITA@G...	1009	2009	HUMAN RE...
44420	RAKESH K...	MALE	44 DL ROAD	19/03/1995	DELHI	B.SC	22/05/2024	16:00:00	110002	9765432190	RAKESH@...	1009	2009	PRODUCTI...
51843	KARAN SIN...	MALE	90 MN STR...	21/02/1999	JAIPUR	M.SC	16/12/2022	12:00:00	302001	9876012345	KARAN@G...	1005	20005	FINANCE
70103	MEENA RE...	FEMALE	11 AP ROAD	13/03/1996	HYDERBAD	B.COM	14/05/2023	16:00:00	500001	9867023456	MEENA@G...	1006	2006	PRODUCTI...
83912	MUSKAN	FEMALE	SHASTRI N...	15/07/1999	KANPUR	B.COM	03/06/2024	09:00:00	208005	55555555555	MUSKAN@...	12345	342	SALES
83957	LALIT	MALE	VIJAY NAGAR	16/10/1999	KANPUR	B.COM	05/06/2024	12:00:00	1234	88888	LAIT@GMAI...	12345	555	FINANCE
89989	AMIT SHAR...	MALE	12 MG ROAD	15/04/1985	MUMBAI	B.SC	01/01/2023	09:00:00	40001	9820123456	AMIT@GMA...	1001	2001	SALES
90918	VANSHIKA	FEMALE	SHASTRI N...	14/10/2004	KANPUR	BCA	03/06/2024	09:00:00	208005	9555190328	VANSHIKA...		12345	INFORMATI...

ADD SALARY STRUCTURE:-

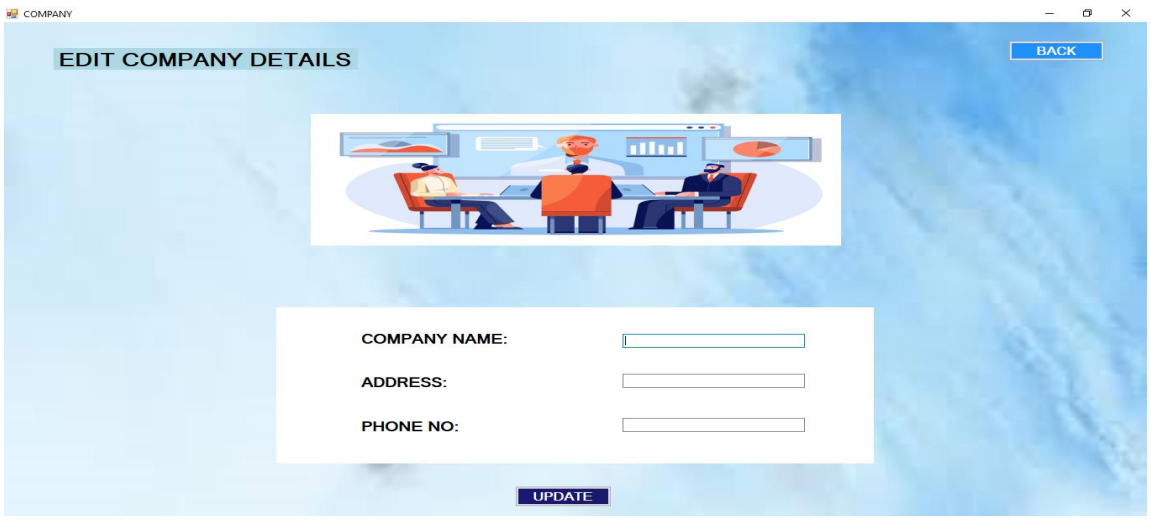
COMPENSATION FRAMEWORK:-



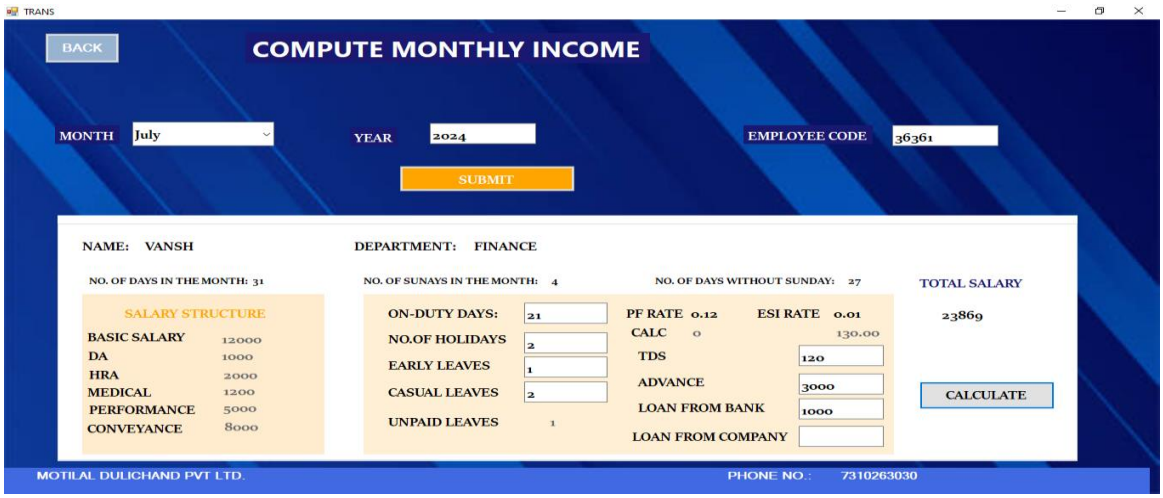
SALARY DETAILS:-

EMPLOYEE_CODE	DEPARTMENT	DESIGNATION	BASIC_SALARY	DA	HRA	MEDICAL	PERFORMANCE	CONVEYANCE
36361	FINANCE	ASSISTANT	12000	1000	2000	1200	5000	8000
83957	FINANCE	SALESMAN	12000	4000	5500	440	300	1000
27747	HUMAN RESOURCE	JUNIOR ASSITANT	32000	3320	5500	1230	6780	4500
89989	SALES	SALES MANAGER	50000	10000	12000	3000	2000	5000
11133	INFORMATION AND T...	SOFTWARE MANAGER	45000	6000	11000	2500	2000	4500
30223	PRODUCTION	AREA SALES MANAG...	60000	12000	14000	4000	3000	6000
27770	INFORMATION AND T...	REGIONAL MANGER ...	70000	14000	16000	4500	3500	7000
70103	PRODUCTION	SALES REPRENTATIVE	12000	2900	5000	2000	1000	1500
17265	HUMAN RESOURCE	SALES EXECUTIVE	30000	8000	2000	1500	1000	3000
51843	FINANCE	SALES OFFICER	25000	5000	7000	1500	1000	2500
34002	SALES	SALES LEAD	40000	8000	10000	2000	1800	4000
40356	HUMAN RESOURCE	SALES ANAYLYST	35000	7000	9000	2000	1500	3500
44420	PRODUCTION	TERRITORY MANAGER	55000	11000	13000	3500	2500	5500

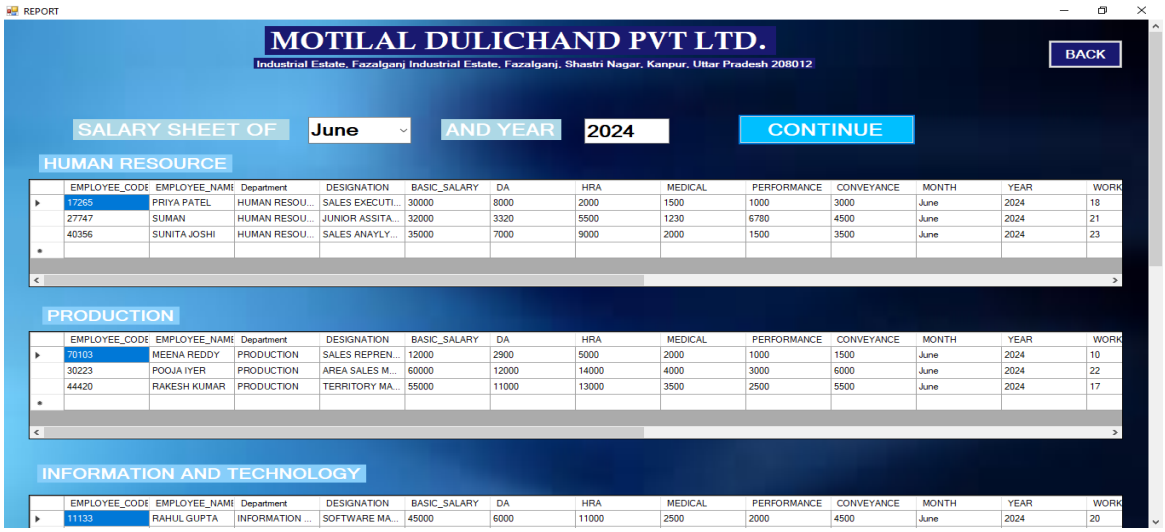
COMPANY DETAILS:-



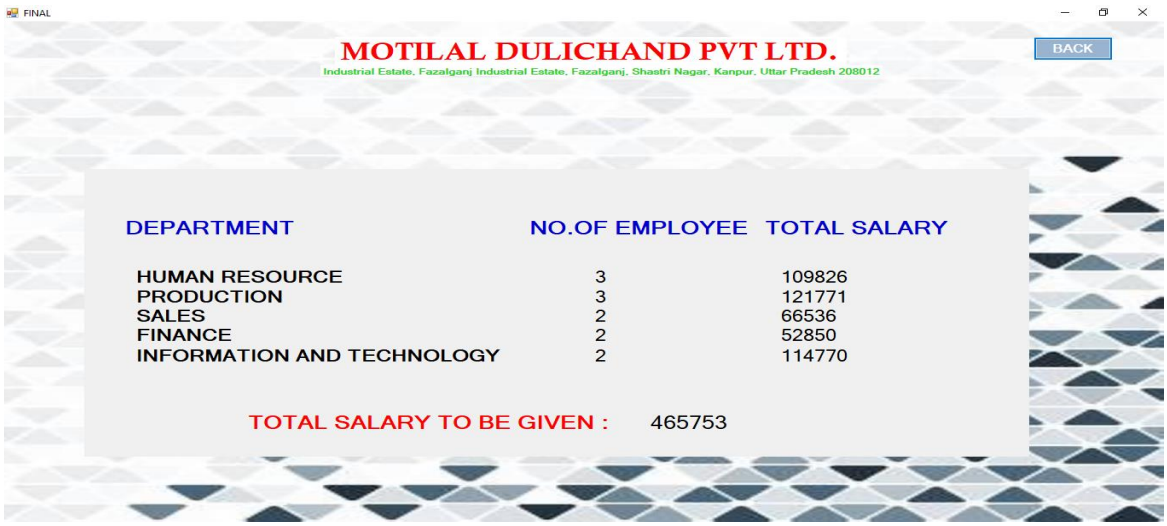
TRANSACTION:-



REPORT:-



SUMMARY:-



CODE IMPLEMENTATION

Code implementation refers to the process of writing the actual code that brings a software design to life. It involves translating design specifications and requirements into a functional program using programming languages. This includes developing the frontend and backend components, integrating them, performing initial testing, and optimizing for efficiency.

Here is the code for all the forms in the payroll system:

1-FORM1:-

```
Imports System.Data.SqlClient

Public Class Form1

    ' Define the connection string (update with your actual database details)
    Private connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ' Call the method to load company details when the form loads
        LoadCompanyDetails()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Hide()
        LOGIN.Show()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Dim result As DialogResult = MessageBox.Show("Are you sure you want to exit the application?",
                                                     "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
        ' If the user confirms, exit the application
        If result = DialogResult.Yes Then
            Application.Exit()
        End If
    End Sub

    Private Sub LoadCompanyDetails()
        ' SQL query to get the company name and address
        Dim query As String = "SELECT [COMPANY NAME], [ADDRESS] FROM [dbo].[COMPANY]"

        ' Create a connection and command object
        Using connection As New SqlConnection(connectionString),
              command As New SqlCommand(query, connection)
            Try
                ' Open the connection
                connection.Open()

                ' Execute the command and get the data reader
```

```
            Using reader As SqlDataReader = command.ExecuteReader()
                ' Check if there is a row to read
                If reader.Read() Then
                    ' Get the name and address from the reader
                    Dim companyName As String = reader("COMPANY NAME").ToString()
                    Dim companyAddress As String = reader("ADDRESS").ToString()

                    ' Set the labels' text to the company details
                    Label2.Text = companyName
                    Label3.Text = companyAddress
                Else
                    ' Handle the case where no data is returned
                    MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                End If
            End Using

        Catch ex As Exception
            ' Handle any errors that might have occurred
            MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Using
  End Sub
End Class
```

2-LOGIN:-

```
Imports System.Data.SqlClient
Imports System.Windows.Forms.VisualStyles.VisualStyleElement.Button

Public Class LOGIN
    Private connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    Private Sub LOGIN_Load_1(sender As Object, e As EventArgs) Handles MyBase.Load
        If CheckBox1.Checked = False Then
            TextBox2.UseSystemPasswordChar = True
        End If
        LoadCompanyDetails()
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim username As String = TextBox1.Text.Trim()
        Dim password As String = TextBox2.Text.Trim()
        If String.IsNullOrEmpty(username) OrElse String.IsNullOrEmpty(password) Then
            MessageBox.Show("Please enter both username and password.", "warning",
            MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Return
        End If
        If TextBox1.Text = ("VANSHIKA") And TextBox2.Text = ("123456") Then
            HOME.Show()
            Me.Hide()
        Else
            MessageBox.Show("Invalid username or password.", "warning", MessageBoxButtons.OK,
            MessageBoxIcon.Warning)
        End If
    End Sub

    Private Sub CheckBox1_CheckedChanged_1(sender As Object, e As EventArgs) Handles CheckBox1.CheckedChanged
        If CheckBox1.Checked = True Then
            TextBox2.UseSystemPasswordChar = False
        ElseIf CheckBox1.Checked = False Then
            TextBox2.UseSystemPasswordChar = True
        End If
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Me.Close()
        Form1.Show()
    End Sub


    Private Sub LoadCompanyDetails()
        ' SQL query to get the company name and address
        Dim query As String = "SELECT [COMPANY NAME] FROM [dbo].[COMPANY]"

        ' Create a connection and command object
        Using connection As New SqlConnection(connectionString),
              command As New SqlCommand(query, connection)
            Try
```

```
            ' Open the connection
            connection.Open()

            ' Execute the command and get the data reader
            Using reader As SqlDataReader = command.ExecuteReader()
                ' Check if there is a row to read
                If reader.Read() Then
                    ' Get the name and address from the reader
                    Dim companyName As String = reader("COMPANY NAME").ToString()


                    ' Set the labels' text to the company details
                    Label4.Text = companyName

                Else
                    ' Handle the case where no data is returned
                    MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                End If
            End Using

        Catch ex As Exception
            ' Handle any errors that might have occurred
            MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Using
  End Sub
End Class
```

3-HOME PAGE:-

```
Imports System.Data.SqlClient

Public Class HOME
    Private connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
    Private Sub LOGOUTToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles LOGOUTToolStripMenuItem.Click
        Dim result As DialogResult
        result = MessageBox.Show("ARE YOU SURE YOU WANT TO LOGOUT", "Confirmation", MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            Me.Close()
            Form1.Show()
        End If
    End Sub

    Private Sub ADDNEWEMPLOYEEToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ADDNEWEMPLOYEEToolStripMenuItem.Click
        Me.Close()
        ADD.Show()
    End Sub

    Private Sub SHOWEMPLOYEEDETAILSToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles SHOWEMPLOYEEDETAILSToolStripMenuItem.Click
        Me.Close()
        SHOWW.Show()
    End Sub

    Private Sub ADDSALARYSTRUCTURETToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ADDSALARYSTRUCTURETToolStripMenuItem.Click
        Me.Close()
        SEARCH.Show()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        LEARN.Show()
    End Sub

    Private Sub COMPENSATIONFRAMEWORKToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles COMPENSATIONFRAMEWORKToolStripMenuItem.Click
        Me.Close()
        PARAMETER.Show()
    End Sub

    Private Sub TRANSACTIONToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles TRANSACTIONToolStripMenuItem.Click
        Me.Close()
        TRANS.Show()
    End Sub
    Private Sub REPORToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles REPORToolStripMenuItem.Click
        Me.Close()
```

```
    REPORT.Show()
  End Sub

  Private Sub EDITSALARYSTRUCTURETOolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles EDITSALARYSTRUCTURETOolStripMenuItem.Click
    Me.Close()
    SALARY_EDIT.Show()
  End Sub

  Private Sub CompanyBrandingToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles CompanyBrandingToolStripMenuItem.Click
    Me.Close()
    COMPANY.Show()
  End Sub

  Private Sub HOME_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    LoadCompanyDetails()
  End Sub
  Private Sub LoadCompanyDetails()
    ' SQL query to get the company name and address
    Dim query As String = "SELECT [COMPANY NAME], [PHONE] FROM [dbo].[COMPANY]"

    ' Create a connection and command object
    Using connection As New SqlConnection(connectionString),
        command As New SqlCommand(query, connection)
      Try
        ' Open the connection
        connection.Open()

        ' Execute the command and get the data reader
        Using reader As SqlDataReader = command.ExecuteReader()
          ' Check if there is a row to read
          If reader.Read() Then
            ' Get the name and address from the reader
            Dim companyName As String = reader("COMPANY NAME").ToString()
            Dim companyPHONE As String = reader("PHONE").ToString()

            ' Set the labels' text to the company details
            Label2.Text = companyName
            Label3.Text = companyPHONE
          Else
            ' Handle the case where no data is returned
            MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
          End If
        End Using

      Catch ex As Exception
        ' Handle any errors that might have occurred
        MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
      End Try
    End Using
  End Sub
End Class
```

4-ADD EMPLOYEE:-

```
Imports System.Data.SqlClient

Public Class ADD
    ' Define a function to generate a unique 5-digit employee code
    Private connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
    Private Function GenerateEmployeeCode() As String
        Dim rnd As New Random()
        Dim code As String
        Do
            code = rnd.Next(10000, 99999).ToString() ' Generate a random 5-digit number
        Loop While IsEmployeeCodeExists(code)
        Return code
    End Function

    ' Check if the generated employee code already exists in the database
    Private Function IsEmployeeCodeExists(code As String) As Boolean
        Dim con As SqlConnection = New SqlConnection("Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False")
        Dim cmd As SqlCommand = New SqlCommand("SELECT COUNT(*) FROM EMPLOYEE WHERE EMPLOYEE_CODE = @Code", con)
        cmd.Parameters.AddWithValue("@Code", code)
        con.Open()
        Dim count As Integer = Convert.ToInt32(cmd.ExecuteScalar())
        con.Close()
        Return count > 0
    End Function

    Private Sub ADD_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ' Populate the ComboBox with shift timings
        ComboBox3.Items.Add("9-12")
        ComboBox3.Items.Add("12-4")
        ComboBox3.Items.Add("4-7")
        LoadCompanyDetails()

        ' Add event handlers for all input controls
        AddHandler TextBox1.TextChanged, AddressOf CheckFields
        AddHandler ComboBox1.SelectedIndexChanged, AddressOf CheckFields
        AddHandler TextBox3.TextChanged, AddressOf CheckFields
        AddHandler DateTimePicker1.ValueChanged, AddressOf CheckFields
        AddHandler TextBox4.TextChanged, AddressOf CheckFields
        AddHandler TextBox8.TextChanged, AddressOf CheckFields
        AddHandler DateTimePicker2.ValueChanged, AddressOf CheckFields
        AddHandler ComboBox3.SelectedIndexChanged, AddressOf CheckFields
        AddHandler TextBox5.TextChanged, AddressOf CheckFields
        AddHandler TextBox9.TextChanged, AddressOf CheckFields
        AddHandler TextBox10.TextChanged, AddressOf CheckFields
        AddHandler ComboBox2.SelectedIndexChanged, AddressOf CheckFields
        AddHandler TextBox11.TextChanged, AddressOf CheckFields
        AddHandler ComboBox4.SelectedIndexChanged, AddressOf CheckFields

        ' Initially check fields
        CheckFields()
```

```
    End Sub

    Private Sub CheckFields()
        ' Check if all required fields are filled
        If Not String.IsNullOrEmpty(TextBox1.Text) AndAlso
           ComboBox1.SelectedIndex <> -1 AndAlso
           Not String.IsNullOrEmpty(TextBox3.Text) AndAlso
           Not String.IsNullOrEmpty(TextBox4.Text) AndAlso
           Not String.IsNullOrEmpty(TextBox8.Text) AndAlso
           ComboBox3.SelectedIndex <> -1 AndAlso
           Not String.IsNullOrEmpty(TextBox5.Text) AndAlso
           Not String.IsNullOrEmpty(TextBox9.Text) AndAlso
           Not String.IsNullOrEmpty(TextBox10.Text) AndAlso
           ComboBox2.SelectedIndex <> -1 AndAlso
           Not String.IsNullOrEmpty(TextBox11.Text) AndAlso
           ComboBox4.SelectedIndex <> -1 Then
            Button2.Enabled = True
        Else
            Button2.Enabled = False
        End If
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        HOME.Show()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Dim empCode As String = GenerateEmployeeCode()
        Dim empName As String = TextBox1.Text
        Dim Department As String = ComboBox4.SelectedItem.ToString()
        Dim con As SqlConnection = New SqlConnection("Data Source=DESKTOP-
DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False")

        Dim cmd As SqlCommand = New SqlCommand("INSERT INTO [dbo].[EMPLOYEE1]
           ([EMPLOYEE_CODE]
           ,[EMPLOYEE_NAME]
           ,[GENDER]
           ,[ADDRESS]
           ,[DOB]
           ,[CITY]
           ,[QUALIFICATION]
           ,[DOJ]
           ,[SHIFT]
           ,[PINCODE]
           ,[PHONE]
           ,[EMAIL]
           ,[PF_NUMBER]
           ,[ESI_NUMBER]
           ,[DEPARTMENT])
         VALUES
        (@EmpCode, @EmpName, @Gender, @Address, @DOB, @City, @Qualification, @DOJ,
@Shift, @Pincode, @Phone, @Email, @PFNumber, @ESINumber,@DEPARTMENT)", con)
        cmd.Parameters.AddWithValue("@EmpCode", empCode)
        cmd.Parameters.AddWithValue("@EmpName", TextBox1.Text)
```

```
        cmd.Parameters.AddWithValue("@Gender", ComboBox1.SelectedItem.ToString())
        cmd.Parameters.AddWithValue("@Address", TextBox3.Text)
        cmd.Parameters.AddWithValue("@DOB", DateTimePicker1.Value)
        cmd.Parameters.AddWithValue("@City", TextBox4.Text)
        cmd.Parameters.AddWithValue("@Qualification", TextBox8.Text)
        cmd.Parameters.AddWithValue("@DOJ", DateTimePicker2.Value)
        cmd.Parameters.AddWithValue("@Shift", TimeSpan.Parse(GetSelectedShiftTime()))
        cmd.Parameters.AddWithValue("@Pincode", TextBox5.Text)
        cmd.Parameters.AddWithValue("@Phone", TextBox9.Text)
        cmd.Parameters.AddWithValue("@Email", TextBox10.Text)
        If ComboBox2.SelectedItem IsNot Nothing Then
            If ComboBox2.SelectedItem.ToString() = "YES" Then
                ' Assuming cmd is your SQL command object
                cmd.Parameters.AddWithValue("@PFNumber", TextBox2.Text)
            Else
                ' Assuming cmd is your SQL command object
                cmd.Parameters.AddWithValue("@PFNumber", DBNull.Value)
            End If
        End If
        ' You can set this value according to your logic
        cmd.Parameters.AddWithValue("@ESINumber", TextBox11.Text)
        cmd.Parameters.AddWithValue("@DEPARTMENT",   ComboBox4.SelectedItem.ToString())   '
You can set this value according to your logic
        con.Open()
        cmd.ExecuteNonQuery()
        Dim salaryForm As New SAL(empCode, empName, Department)
        con.Close()
        MessageBox.Show("Employee added successfully")
        salaryForm.Show()
        Me.Close()
    End Sub

    Private Function GetSelectedShiftTime() As String
        ' Retrieve the selected shift timing from the ComboBox
        Select Case ComboBox3.SelectedItem.ToString()
            Case "9-12"
                Return "09:00:00" ' Start time of the shift
            Case "12-4"
                Return "12:00:00"
            Case "4-7"
                Return "16:00:00"
            Case Else
                Return "00:00:00" ' Default value
        End Select
    End Function

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Me.Close()
        HOME.Show()
    End Sub
    Private Sub LoadCompanyDetails()
        ' SQL query to get the company name and address
        Dim query As String = "SELECT [COMPANY NAME], [PHONE] FROM [dbo].[COMPANY]"

        ' Create a connection and command object
```

```
        Using connection As New SqlConnection(connectionString),
            command As New SqlCommand(query, connection)
          Try
            ' Open the connection
            connection.Open()

            ' Execute the command and get the data reader
            Using reader As SqlDataReader = command.ExecuteReader()
              ' Check if there is a row to read
              If reader.Read() Then
                ' Get the name and address from the reader
                Dim companyName As String = reader("COMPANY NAME").ToString()
                Dim companyPHONE As String = reader("PHONE").ToString()

                ' Set the labels' text to the company details
                Label19.Text = companyName
                Label18.Text = companyPHONE
              Else
                ' Handle the case where no data is returned
                MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
              End If
            End Using

          Catch ex As Exception
            ' Handle any errors that might have occurred
            MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
          End Try
        End Using
    End Sub

End Class
```

5-SALARY STRUCTURE:-

```
Imports System.Data.SqlClient
Imports System.Reflection.Emit

Public Class SAL
   Private empCode As String
   Private empName As String
   Private Department As String
   Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
   Public Sub New(empCode As String, empName As String, Department As String)
      InitializeComponent()
      Me.empCode = empCode
      Me.empName = empName
      Me.Department = Department
      Label12.Text = empCode
      Label13.Text = empName
      Label14.Text = Department
   End Sub
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
      Try
         Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

         Using con As New SqlConnection(connectionString)
            con.Open()
            Dim sql As String = "INSERT INTO [dbo].[SALARY] ([EMPLOYEE_CODE], [DEPARTMENT], [DESIGNATION], [BASIC_SALARY], [DA], [HRA], [MEDICAL], [PERFORMANCE], [CONVEYANCE]) VALUES (@EmpCode, @Department, @Designation, @BasicSalary, @DA, @HRA, @Medical, @Performance, @Conveyance)"

            Using cmd As New SqlCommand(sql, con)
               cmd.Parameters.AddWithValue("@EmpCode", empCode)
               cmd.Parameters.AddWithValue("@Department", Department)
               cmd.Parameters.AddWithValue("@Designation", TextBox1.Text)
               cmd.Parameters.AddWithValue("@BasicSalary", If(Decimal.TryParse(TextBox2.Text, New Decimal), Decimal.Parse(TextBox2.Text), DBNull.Value))
               cmd.Parameters.AddWithValue("@DA", If(Decimal.TryParse(TextBox3.Text, New Decimal), Decimal.Parse(TextBox3.Text), DBNull.Value))
               cmd.Parameters.AddWithValue("@HRA", If(Decimal.TryParse(TextBox4.Text, New Decimal), Decimal.Parse(TextBox4.Text), DBNull.Value))
               cmd.Parameters.AddWithValue("@Medical", If(Decimal.TryParse(TextBox5.Text, New Decimal), Decimal.Parse(TextBox5.Text), DBNull.Value))
               cmd.Parameters.AddWithValue("@Performance", If(Decimal.TryParse(TextBox6.Text, New Decimal), Decimal.Parse(TextBox6.Text), DBNull.Value))
               cmd.Parameters.AddWithValue("@Conveyance", If(Decimal.TryParse(TextBox7.Text, New Decimal), Decimal.Parse(TextBox7.Text), DBNull.Value))

               cmd.ExecuteNonQuery()
               MessageBox.Show("Salary details added successfully")
               Me.Close()
               HOME.Show()
            End Using
         End Using
```

```
        Catch ex As Exception
            MessageBox.Show("Error inserting salary details: " & ex.Message)
        End Try
    End Sub


    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Me.Close()
        ADD.Show()
    End Sub

    Private Sub SAL_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadCompanyDetails()
    End Sub
    Private Sub LoadCompanyDetails()
        ' SQL query to get the company name and address
        Dim query As String = "SELECT [COMPANY NAME], [PHONE] FROM [dbo].[COMPANY]"

        ' Create a connection and command object
        Using connection As New SqlConnection(connectionString),
              command As New SqlCommand(query, connection)
            Try
                ' Open the connection
                connection.Open()

                ' Execute the command and get the data reader
                Using reader As SqlDataReader = command.ExecuteReader()
                    ' Check if there is a row to read
                    If reader.Read() Then
                        ' Get the name and address from the reader
                        Dim companyName As String = reader("COMPANY NAME").ToString()
                        Dim companyPHONE As String = reader("PHONE").ToString()

                        ' Set the labels' text to the company details
                        Label18.Text = companyName
                        Label17.Text = companyPHONE
                    Else
                        ' Handle the case where no data is returned
                        MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                    End If
                End Using

            Catch ex As Exception
                ' Handle any errors that might have occurred
                MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End Using
    End Sub

End Class
```

6-EDIT EMPLOYEE:-

```
Imports System.Data.SqlClient

Public Class SHOWW
    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        HOME.Show()
    End Sub

    Private Sub SHOWW_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadDataIntoDataGridView()
        DataGridView1.AllowUserToAddRows = False
        DataGridView1.AllowUserToDeleteRows = False
        DataGridView1.ReadOnly = False
        LoadCompanyDetails()
    End Sub

    Private Sub LoadDataIntoDataGridView()
        Try
            Dim connection As New SqlConnection(connectionString)
            connection.Open()
            Dim query As String = "SELECT * FROM EMPLOYEE1"
            Dim adapter As New SqlDataAdapter(query, connection)
            Dim commandBuilder As New SqlCommandBuilder(adapter)
            Dim dataSet As New DataSet()
            adapter.Fill(dataSet, "EMPLOYEE")
            DataGridView1.DataSource = dataSet.Tables("EMPLOYEE")
            connection.Close()
        Catch ex As Exception
            MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ' Update the selected row
        If DataGridView1.SelectedRows.Count > 0 Then
            Try
                Dim row As DataGridViewRow = DataGridView1.SelectedRows(0)
                Dim employeeCode As String = row.Cells("EMPLOYEE_CODE").Value.ToString() ' Assuming "EMPLOYEE_CODE" is the unique column

                Dim updateQuery As String = "UPDATE EMPLOYEE1 SET "
                Dim parameters As New List(Of SqlParameter)

                For Each cell As DataGridViewCell In row.Cells
                    If cell.OwningColumn.Name <> "EMPLOYEE_CODE" Then ' Skip the unique column
                        updateQuery &= cell.OwningColumn.Name & " = @" & cell.OwningColumn.Name & ", "
                        parameters.Add(New SqlParameter("@" & cell.OwningColumn.Name, cell.Value))
                    End If
```

```
            Next

            ' Remove the last comma and add the WHERE clause
            updateQuery = updateQuery.TrimEnd(","c, " "c) & " WHERE EMPLOYEE_CODE =
@EMPLOYEE_CODE"
            parameters.Add(New SqlParameter("@EMPLOYEE_CODE", employeeCode))

            Dim connection As New SqlConnection(connectionString)
            Dim command As New SqlCommand(updateQuery, connection)
            command.Parameters.AddRange(parameters.ToArray())
            connection.Open()
            command.ExecuteNonQuery()
            connection.Close()
            MessageBox.Show("Record updated successfully", "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information)
          Catch ex As Exception
            MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
          End Try
        Else
          MessageBox.Show("Please select a row to update", "Warning", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
        End If
    End Sub


    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
      ' Delete the selected row
      If DataGridView1.SelectedRows.Count > 0 Then
        Try
          Dim row As DataGridViewRow = DataGridView1.SelectedRows(0)
          Dim employeeCode As String = row.Cells("EMPLOYEE_CODE").Value.ToString() '
Assuming "EMPLOYEE_CODE" is the unique column

          Dim connection As New SqlConnection(connectionString)
          Dim command As New SqlCommand("DELETE FROM EMPLOYEE1 WHERE
EMPLOYEE_CODE = @EMPLOYEE_CODE", connection)
          command.Parameters.AddWithValue("@EMPLOYEE_CODE", employeeCode)
          connection.Open()
          command.ExecuteNonQuery()
          connection.Close()
          DataGridView1.Rows.Remove(row)
          MessageBox.Show("Record deleted successfully", "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Catch ex As Exception
          MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        End Try
      Else
        MessageBox.Show("Please select a row to delete", "Warning", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
      End If
    End Sub
    Private Sub LoadCompanyDetails()
      ' SQL query to get the company name and address
```

```
        Dim query As String = "SELECT [COMPANY NAME], [PHONE] FROM [dbo].[COMPANY]"

        ' Create a connection and command object
        Using connection As New SqlConnection(connectionString),
              command As New SqlCommand(query, connection)
            Try
                ' Open the connection
                connection.Open()

                ' Execute the command and get the data reader
                Using reader As SqlDataReader = command.ExecuteReader()
                    ' Check if there is a row to read
                    If reader.Read() Then
                        ' Get the name and address from the reader
                        Dim companyName As String = reader("COMPANY NAME").ToString()
                        Dim companyPHONE As String = reader("PHONE").ToString()

                        ' Set the labels' text to the company details
                        Label2.Text = companyName
                        Label3.Text = companyPHONE
                    Else
                        ' Handle the case where no data is returned
                        MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                    End If
                End Using

            Catch ex As Exception
                ' Handle any errors that might have occurred
                MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End Using
    End Sub
End Class
```

7-ADD SALARY STRUCTURE:-

```
Imports System.Data.SqlClient


Public Class SEARCH

    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click

        Dim employeeCode As String = TextBox8.Text.Trim()


        If IsNumeric(employeeCode) Then

            If EmployeeCodeExists(Convert.ToInt32(employeeCode)) Then

                GroupBox1.Visible = True

                Panel2.Visible = True

                Panel1.Visible = False

                FetchEmployeeData(employeeCode)

                ' Fetch department and add data to the salary table

            Else

                MessageBox.Show("Employee code does not exist.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

            End If

        Else

            MessageBox.Show("Please enter a valid numeric employee code.", "Invalid Input", MessageBoxButtons.OK, MessageBoxIcon.Warning)

        End If

    End Sub

    Private Sub FetchEmployeeData(employeeCode As String)

        Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

        Dim query As String = "SELECT EMPLOYEE_CODE,EMPLOYEE_NAME, DEPARTMENT FROM EMPLOYEE1 WHERE EMPLOYEE_CODE = @EmployeeCode"


        Using connection As New SqlConnection(connectionString)

            Using command As New SqlCommand(query, connection)

                command.Parameters.AddWithValue("@EmployeeCode", employeeCode)
```

```
            connection.Open()
            Using reader As SqlDataReader = command.ExecuteReader()
                If reader.HasRows Then
                    reader.Read()
                    Label11.Text = reader("EMPLOYEE_CODE").ToString()
                    Label13.Text = reader("EMPLOYEE_NAME").ToString()
                    Label15.Text = reader("DEPARTMENT").ToString()


                    GroupBox1.Visible = True
                Else
                    MessageBox.Show("Employee not found.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
                    GroupBox1.Visible = False
                End If
            End Using
        End Using
    End Using
End Sub


Private Function EmployeeCodeExists(employeeCode As Integer) As Boolean
    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
    Dim query As String = "SELECT COUNT(*) FROM EMPLOYEE1 WHERE EMPLOYEE_CODE = @EmployeeCode"
    Using connection As New SqlConnection(connectionString)
        Using command As New SqlCommand(query, connection)
            command.Parameters.AddWithValue("@EmployeeCode", employeeCode)
            connection.Open()
            Dim count As Integer = Convert.ToInt32(command.ExecuteScalar())
            Return count > 0
        End Using
    End Using
End Function
Private Function GetEmployeeDepartment(employeeCode As Integer) As String
```

```
    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    Dim query As String = "SELECT DEPARTMENT FROM EMPLOYEE1 WHERE EMPLOYEE_CODE = @EmployeeCode"

    Using connection As New SqlConnection(connectionString)

        Using command As New SqlCommand(query, connection)

            command.Parameters.AddWithValue("@EmployeeCode", employeeCode)

            connection.Open()

            Dim department As Object = command.ExecuteScalar()

            If department IsNot Nothing Then

                Return department.ToString()

            Else

                Return String.Empty

            End If

        End Using

    End Using

End Function

Private Function SalaryDataExists(employeeCode As Integer) As Boolean

    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    Dim query As String = "SELECT COUNT(*) FROM SALARY WHERE EMPLOYEE_CODE = @EmployeeCode"

    Using connection As New SqlConnection(connectionString)

        Using command As New SqlCommand(query, connection)

            command.Parameters.AddWithValue("@EmployeeCode", employeeCode)

            connection.Open()

            Dim count As Integer = Convert.ToInt32(command.ExecuteScalar())

            Return count > 0

        End Using

    End Using

End Function

Private Sub AddSalaryData(empCode As Integer, Department As String, Designation As String, BasicSalary As String, DA As String, HRA As String, Medical As String, Performance As String, Conveyance As String)

    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
```

```
        Try
            Using con As New SqlConnection(connectionString)
                con.Open()
            Dim sql As String = "INSERT INTO [dbo].[SALARY] ([EMPLOYEE_CODE], [DEPARTMENT], [DESIGNATION], [BASIC_SALARY], [DA], [HRA], [MEDICAL], [PERFORMANCE], [CONVEYANCE]) VALUES (@EmpCode, @Department, @Designation, @BasicSalary, @DA, @HRA, @Medical, @Performance, @Conveyance)"

                Using cmd As New SqlCommand(sql, con)
                    cmd.Parameters.AddWithValue("@EmpCode", empCode)
                    cmd.Parameters.AddWithValue("@Department", Department)
                    cmd.Parameters.AddWithValue("@Designation", Designation)
                    cmd.Parameters.AddWithValue("@BasicSalary", If(Decimal.TryParse(BasicSalary, New Decimal), Decimal.Parse(BasicSalary), DBNull.Value))
                    cmd.Parameters.AddWithValue("@DA", If(Decimal.TryParse(DA, New Decimal), Decimal.Parse(DA), DBNull.Value))
                    cmd.Parameters.AddWithValue("@HRA", If(Decimal.TryParse(HRA, New Decimal), Decimal.Parse(HRA), DBNull.Value))
                    cmd.Parameters.AddWithValue("@Medical", If(Decimal.TryParse(Medical, New Decimal), Decimal.Parse(Medical), DBNull.Value))
                    cmd.Parameters.AddWithValue("@Performance", If(Decimal.TryParse(Performance, New Decimal), Decimal.Parse(Performance), DBNull.Value))
                    cmd.Parameters.AddWithValue("@Conveyance", If(Decimal.TryParse(Conveyance, New Decimal), Decimal.Parse(Conveyance), DBNull.Value))

                    cmd.ExecuteNonQuery()
                    MessageBox.Show("Salary details added successfully")
                End Using
            End Using
        Catch ex As Exception
            MessageBox.Show("Error inserting salary details: " & ex.Message)
        End Try
    End Sub

    Private Sub SEARCH_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        GroupBox1.Visible = False
        Panel2.Visible = False
```

```
        If GroupBox1.Visible = True Then
            Panel1.Visible = False
        Else
            Panel1.Visible = True
        End If
        LoadCompanyDetails()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim employeeCode As String = TextBox8.Text.Trim()
        If IsNumeric(employeeCode) Then
            Dim empCode As Integer = Convert.ToInt32(employeeCode)
            If Not SalaryDataExists(empCode) Then
                Dim department As String = GetEmployeeDepartment(empCode)
                If Not String.IsNullOrEmpty(department) Then
                    AddSalaryData(empCode, department, TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text, TextBox6.Text, TextBox7.Text)
                Else
                    MessageBox.Show("Department not found for the given employee code.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
                End If
            Else
                MessageBox.Show("Salary data already exists for the given employee code.", "Data Exists", MessageBoxButtons.OK, MessageBoxIcon.Information)
            End If
        Else
            MessageBox.Show("Please enter a valid numeric employee code.", "Invalid Input", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Me.Close()
        HOME.Show()
    End Sub
```

```
    Private Sub LoadCompanyDetails()
        ' SQL query to get the company name and address
        Dim query As String = "SELECT [COMPANY NAME], [PHONE] FROM [dbo].[COMPANY]"
        ' Create a connection and command object
        Using connection As New SqlConnection(connectionString),
            command As New SqlCommand(query, connection)
            Try
                ' Open the connection
                connection.Open()
                ' Execute the command and get the data reader
                Using reader As SqlDataReader = command.ExecuteReader()
                    ' Check if there is a row to read
                    If reader.Read() Then
                        ' Get the name and address from the reader
                        Dim companyName As String = reader("COMPANY NAME").ToString()
                        Dim companyPHONE As String = reader("PHONE").ToString()
                        ' Set the labels' text to the company details
                        Label18.Text = companyName
                        Label17.Text = companyPHONE
                    Else
                        ' Handle the case where no data is returned
                        MessageBox.Show("No company details found.", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information)
                    End If
                End Using
            Catch ex As Exception
                ' Handle any errors that might have occurred
                MessageBox.Show("An error occurred while fetching company details: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End Using
    End Sub

End Class
```

8-COMPENSATION FRAMEWORK:-

```
Imports System.Data.SqlClient

Public Class PARAMETER
    Private connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Try
            Using con As New SqlConnection(connectionString)
                con.Open()
                Dim sql As String = "UPDATE [dbo].[PARAMETER] SET
                                    [PF LIMIT] = @PF_Limit,
                                    [PF EMPLOYEE RATE] = @PF_Employee_Rate,
                                    [PF EMPLOYER RATE] = @PF_Employer_Rate,
                                    [ESI LIMIT] = @ESI_Limit,
                                    [ESI EMPLOYEE RATE] = @ESI_Employee_Rate,
                                    [ESI EMPLOYER RATE] = @ESI_Employer_Rate"

                Using cmd As New SqlCommand(sql, con)
                    ' Parsing and adding parameters
                    cmd.Parameters.AddWithValue("@PF_Limit", If(Decimal.TryParse(TextBox1.Text, New Decimal), Decimal.Parse(TextBox1.Text), DBNull.Value))
                    cmd.Parameters.AddWithValue("@PF_Employee_Rate", If(Decimal.TryParse(TextBox2.Text, New Decimal), Decimal.Parse(TextBox2.Text), DBNull.Value))
                    cmd.Parameters.AddWithValue("@PF_Employer_Rate", If(Decimal.TryParse(TextBox3.Text, New Decimal), Decimal.Parse(TextBox3.Text), DBNull.Value))

                    cmd.Parameters.AddWithValue("@ESI_Limit", If(Decimal.TryParse(TextBox4.Text, New Decimal), Decimal.Parse(TextBox4.Text), DBNull.Value))
                    cmd.Parameters.AddWithValue("@ESI_Employee_Rate", If(Decimal.TryParse(TextBox5.Text, New Decimal), Decimal.Parse(TextBox5.Text), DBNull.Value))
                    cmd.Parameters.AddWithValue("@ESI_Employer_Rate", If(Decimal.TryParse(TextBox6.Text, New Decimal), Decimal.Parse(TextBox6.Text), DBNull.Value))

                    ' Execute the update command
                    cmd.ExecuteNonQuery()
                    MessageBox.Show("Compensation parameters updated successfully")
                End Using
            End Using
        Catch ex As Exception
            MessageBox.Show("Error updating compensation parameters: " & ex.Message)
        End Try
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Me.Close()
        HOME.Show()
    End Sub
```

```
    Private Sub PARAMETER_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadCompanyDetails()
        LoadParameterValues()
    End Sub

    Private Sub LoadCompanyDetails()
        ' SQL query to get the company name and address
        Dim query As String = "SELECT [COMPANY NAME], [PHONE] FROM [dbo].[COMPANY]"

        ' Create a connection and command object
        Using connection As New SqlConnection(connectionString),
              command As New SqlCommand(query, connection)
            Try
                ' Open the connection
                connection.Open()

                ' Execute the command and get the data reader
                Using reader As SqlDataReader = command.ExecuteReader()
                    ' Check if there is a row to read
                    If reader.Read() Then
                        ' Get the name and address from the reader
                        Dim companyName As String = reader("COMPANY NAME").ToString()
                        Dim companyPHONE As String = reader("PHONE").ToString()

                        ' Set the labels' text to the company details
                        Label10.Text = companyName
                        Label9.Text = companyPHONE
                    Else
                        ' Handle the case where no data is returned
                        MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                    End If
                End Using

            Catch ex As Exception
                ' Handle any errors that might have occurred
                MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End Using
    End Sub

    Private Sub LoadParameterValues()
        ' SQL query to get the parameter values
        Dim query As String = "SELECT [PF LIMIT], [PF EMPLOYEE RATE], [PF EMPLOYER
RATE], [ESI LIMIT], [ESI EMPLOYEE RATE], [ESI EMPLOYER RATE] FROM
[dbo].[PARAMETER]"

        ' Create a connection and command object
        Using connection As New SqlConnection(connectionString),
              command As New SqlCommand(query, connection)
            Try
                ' Open the connection
                connection.Open()
```

```
            ' Execute the command and get the data reader
            Using reader As SqlDataReader = command.ExecuteReader()
                ' Check if there is a row to read
                If reader.Read() Then
                    ' Get the parameter values from the reader
                    TextBox1.Text = reader("PF LIMIT").ToString()
                    TextBox2.Text = reader("PF EMPLOYEE RATE").ToString()
                    TextBox3.Text = reader("PF EMPLOYER RATE").ToString()
                    TextBox4.Text = reader("ESI LIMIT").ToString()
                    TextBox5.Text = reader("ESI EMPLOYEE RATE").ToString()
                    TextBox6.Text = reader("ESI EMPLOYER RATE").ToString()
                Else
                    ' Handle the case where no data is returned
                    MessageBox.Show("No parameter values found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                End If
            End Using

        Catch ex As Exception
            ' Handle any errors that might have occurred
            MessageBox.Show("An error occurred while fetching parameter values: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Using
  End Sub
End Class
```

9-EDIT SALARY STRUCTURE:-

```
Imports System.Data.SqlClient
Imports System.Reflection.Emit

Public Class SALARY_EDIT

    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        HOME.Show()
    End Sub

    Private Sub SHOWW_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadDataIntoDataGridView()
        DataGridView1.AllowUserToAddRows = False
        DataGridView1.AllowUserToDeleteRows = False
        DataGridView1.ReadOnly = False
        LoadCompanyDetails()
    End Sub

    Private Sub LoadDataIntoDataGridView()
        Try
            Dim connection As New SqlConnection(connectionString)
            connection.Open()
            Dim query As String = "SELECT * FROM SALARY"
            Dim adapter As New SqlDataAdapter(query, connection)
            Dim commandBuilder As New SqlCommandBuilder(adapter)
            Dim dataSet As New DataSet()
            adapter.Fill(dataSet, "SALARY")
            DataGridView1.DataSource = dataSet.Tables("SALARY")
            connection.Close()
        Catch ex As Exception
            MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ' Update the selected row
        If DataGridView1.SelectedRows.Count > 0 Then
            Try
                Dim row As DataGridViewRow = DataGridView1.SelectedRows(0)
                Dim employeeCode As String = row.Cells("EMPLOYEE_CODE").Value.ToString() ' Assuming "EMPLOYEE_CODE" is the unique column

                Dim updateQuery As String = "UPDATE SALARY SET "
                Dim parameters As New List(Of SqlParameter)

                For Each cell As DataGridViewCell In row.Cells
                    If cell.OwningColumn.Name <> "EMPLOYEE_CODE" Then ' Skip the unique column
                        updateQuery &= cell.OwningColumn.Name & " = @" & cell.OwningColumn.Name & ", "
```

```
                parameters.Add(New SqlParameter("@" & cell.OwningColumn.Name, cell.Value))
            End If
        Next

        ' Remove the last comma and add the WHERE clause
        updateQuery = updateQuery.TrimEnd(","c, " "c) & " WHERE EMPLOYEE_CODE = @EMPLOYEE_CODE"
        parameters.Add(New SqlParameter("@EMPLOYEE_CODE", employeeCode))

        Dim connection As New SqlConnection(connectionString)
        Dim command As New SqlCommand(updateQuery, connection)
        command.Parameters.AddRange(parameters.ToArray())
        connection.Open()
        command.ExecuteNonQuery()
        connection.Close()
        MessageBox.Show("Record updated successfully", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information)
      Catch ex As Exception
        MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
      End Try
    Else
      MessageBox.Show("Please select a row to update", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning)
    End If
  End Sub

  Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    ' Delete the selected row
    If DataGridView1.SelectedRows.Count > 0 Then
      Try
        Dim row As DataGridViewRow = DataGridView1.SelectedRows(0)
        Dim employeeCode As String = row.Cells("EMPLOYEE_CODE").Value.ToString() ' Assuming "EMPLOYEE_CODE" is the unique column

        Dim connection As New SqlConnection(connectionString)
        Dim command As New SqlCommand("DELETE FROM SALARY WHERE EMPLOYEE_CODE = @EMPLOYEE_CODE", connection)
        command.Parameters.AddWithValue("@EMPLOYEE_CODE", employeeCode)
        connection.Open()
        command.ExecuteNonQuery()
        connection.Close()
        DataGridView1.Rows.Remove(row)
        MessageBox.Show("Record deleted successfully", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information)
      Catch ex As Exception
        MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
      End Try
    Else
      MessageBox.Show("Please select a row to delete", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning)
    End If
  End Sub
  Private Sub LoadCompanyDetails()
```

```
        ' SQL query to get the company name and address
        Dim query As String = "SELECT [COMPANY NAME], [PHONE] FROM [dbo].[COMPANY]"

        ' Create a connection and command object
        Using connection As New SqlConnection(connectionString),
              command As New SqlCommand(query, connection)
            Try
                ' Open the connection
                connection.Open()

                ' Execute the command and get the data reader
                Using reader As SqlDataReader = command.ExecuteReader()
                    ' Check if there is a row to read
                    If reader.Read() Then
                        ' Get the name and address from the reader
                        Dim companyName As String = reader("COMPANY NAME").ToString()
                        Dim companyPHONE As String = reader("PHONE").ToString()

                        ' Set the labels' text to the company details
                        Label2.Text = companyName
                        Label3.Text = companyPHONE
                    Else
                        ' Handle the case where no data is returned
                        MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                    End If
                End Using

            Catch ex As Exception
                ' Handle any errors that might have occurred
                MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End Using
    End Sub
End Class
```

10-COMPANY BRANDING:-

```
Imports System.Data.SqlClient

Public Class COMPANY
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        HOME.Show()

    End Sub



    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ' Define your connection string (update as needed)
        Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

        ' Initialize variables for the new values


        ' Initialize variables for the new values
        Dim newCompanyName As String = TextBox1.Text.Trim()
        Dim newAddress As String = TextBox2.Text.Trim()
        Dim newPhoneNumber As String = TextBox3.Text.Trim()

        ' Initialize variables to hold current values
        Dim currentCompanyName As String = String.Empty
        Dim currentAddress As String = String.Empty
        Dim currentPhoneNumber As String = String.Empty

        ' Retrieve current values from the database
        Using connection As New SqlConnection(connectionString)
            Dim selectCommand As New SqlCommand("SELECT [COMPANY NAME], [ADDRESS], [PHONE] FROM COMPANY", connection)

            connection.Open()
            Using reader As SqlDataReader = selectCommand.ExecuteReader()
                If reader.Read() Then
                    currentCompanyName = reader("COMPANY NAME").ToString()
                    currentAddress = reader("ADDRESS").ToString()
                    currentPhoneNumber = reader("PHONE").ToString()
                End If
            End Using
        End Using

        ' Update the values only if the textboxes are not empty
        If String.IsNullOrEmpty(newCompanyName) Then
            newCompanyName = currentCompanyName
        End If
        If String.IsNullOrEmpty(newAddress) Then
            newAddress = currentAddress
        End If
        If String.IsNullOrEmpty(newPhoneNumber) Then
```

```
            newPhoneNumber = currentPhoneNumber
        End If

        ' Update the database with the new values
        Using connection As New SqlConnection(connectionString)
            Dim updateCommand As New SqlCommand("UPDATE COMPANY SET [COMPANY
NAME] = @CompanyName, [ADDRESS] = @Address, [PHONE] = @Phone", connection)
            updateCommand.Parameters.AddWithValue("@CompanyName", newCompanyName)
            updateCommand.Parameters.AddWithValue("@Address", newAddress)
            updateCommand.Parameters.AddWithValue("@Phone", newPhoneNumber)

            connection.Open()
            updateCommand.ExecuteNonQuery()
        End Using

        ' Show a message indicating success
        MessageBox.Show("Company details updated successfully.")
    End Sub
End Class
```

11-TRANSACTION:-

```
Imports System.Data.SqlClient
Imports System.Globalization

Public Class TRANS
    Private Sub Form_Load()
        ' Populate Month ComboBox
        Label45.Text = "0"
        For i As Integer = 1 To 12
            ComboBox1.Items.Add(MonthName(i))
        Next
        ' Set default values to current month and year
        ComboBox1.SelectedIndex = DateTime.Now.Month - 1
        TextBox1.Text = DateTime.Now.Year.ToString()
    End Sub

    Private Sub TRANS_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        GroupBox1.Visible = False
        Label35.Visible = False
        Form_Load()
        LoadCompanyDetails()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim month As Integer = ComboBox1.SelectedIndex + 1
        Dim year As Integer = Integer.Parse(TextBox1.Text)
        DisplayPFAndESIRates()
        Dim employeeCode As String = TextBox2.Text
        GroupBox1.Visible = True
        If String.IsNullOrEmpty(employeeCode) Then
            MessageBox.Show("Please enter the employee code.")
            Return
        End If
```

```
        ' Fetch and display employee and salary details
        FetchEmployeeDetails(employeeCode)
        FetchSalaryDetails(employeeCode)
        DisplayMonthDetails(month, year)
        UpdatePF()
        UpdateESI()

    End Sub

    Private Sub FetchEmployeeDetails(employeeCode As String)
        Dim connString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
        Dim query As String = "SELECT EMPLOYEE_NAME, Department FROM Employee1 WHERE Employee_Code = @EmployeeCode"

        Using conn As New SqlConnection(connString)
            Using cmd As New SqlCommand(query, conn)
                cmd.Parameters.AddWithValue("@EmployeeCode", employeeCode)
                conn.Open()
                Using reader As SqlDataReader = cmd.ExecuteReader()
                    If reader.Read() Then
                        Label5.Text = reader("EMPLOYEE_NAME").ToString()
                        Label7.Text = reader("Department").ToString()
                    Else
                        MessageBox.Show("Employee not found.")
                    End If
                End Using
            End Using
        End Using
    End Sub

    Private Sub FetchSalaryDetails(employeeCode As String)
        Dim connString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
```

```
    Dim query As String = "SELECT Basic_Salary, DA, HRA, Medical, Performance, Conveyance FROM Salary WHERE Employee_Code = @EmployeeCode"

    Using conn As New SqlConnection(connString)
        Using cmd As New SqlCommand(query, conn)
            cmd.Parameters.AddWithValue("@EmployeeCode", employeeCode)

            conn.Open()
            Using reader As SqlDataReader = cmd.ExecuteReader()
                If reader.Read() Then
                    Label8.Text = reader("Basic_Salary").ToString()
                    Label9.Text = reader("DA").ToString()
                    Label10.Text = reader("HRA").ToString()
                    Label11.Text = reader("Medical").ToString()
                    Label12.Text = reader("Performance").ToString()
                    Label13.Text = reader("Conveyance").ToString()
                Else
                    MessageBox.Show("Salary details not found.")
                End If
            End Using
        End Using
    End Using
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    Me.Close()
    HOME.Show()
End Sub
Private Sub LoadCompanyDetails()
    Dim connString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
    ' SQL query to get the company name and address
    Dim query As String = "SELECT [COMPANY NAME], [PHONE] FROM [dbo].[COMPANY]"
```

```
        ' Create a connection and command object
        Using connection As New SqlConnection(connString),
              command As New SqlCommand(query, connection)
            Try
                ' Open the connection
                connection.Open()

                ' Execute the command and get the data reader
                Using reader As SqlDataReader = command.ExecuteReader()
                    ' Check if there is a row to read
                    If reader.Read() Then
                        ' Get the name and address from the reader
                        Dim companyName As String = reader("COMPANY NAME").ToString()
                        Dim companyPHONE As String = reader("PHONE").ToString()

                        ' Set the labels' text to the company details
                        Label44.Text = companyName
                        Label43.Text = companyPHONE
                    Else
                        ' Handle the case where no data is returned
                        MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                    End If
                End Using

            Catch ex As Exception
                ' Handle any errors that might have occurred
                MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End Using
    End Sub
    Private Sub DisplayMonthDetails(month As String, year As Integer)
        Dim daysInMonth As Integer = DateTime.DaysInMonth(year, month)
```

```
        Dim sundays As Integer = CountSundays(month, year)
        Dim daysWithoutSundays As Integer = daysInMonth - sundays

        Label2.Text = daysInMonth.ToString()
        Label40.Text = sundays.ToString()
        Label41.Text = daysWithoutSundays.ToString()
    End Sub

    Private Function CountSundays(month As String, year As Integer) As Integer
        Dim sundays As Integer = 0
        Dim daysInMonth As Integer = DateTime.DaysInMonth(year, month)

        For day As Integer = 1 To daysInMonth
            Dim currentDate As New DateTime(year, month, day)
            If currentDate.DayOfWeek = DayOfWeek.Sunday Then
                sundays += 1
            End If
        Next

        Return sundays
    End Function
    Private Sub UpdateUnpaidLeave()
        Try
            ' Convert TextBox inputs to numbers
            Dim dutyDay As Integer = If(String.IsNullOrEmpty(TextBox3.Text), 0,
Integer.Parse(TextBox3.Text))
            Dim earlyLeave As Integer = If(String.IsNullOrEmpty(TextBox4.Text), 0,
Integer.Parse(TextBox4.Text))
            Dim casualLeave As Integer = If(String.IsNullOrEmpty(TextBox5.Text), 0,
Integer.Parse(TextBox5.Text))
            Dim holiday As Integer = If(String.IsNullOrEmpty(TextBox6.Text), 0,
Integer.Parse(TextBox6.Text))
            Dim deductedValue As Integer = If(String.IsNullOrEmpty(Label41.Text), 0,
Integer.Parse(Label41.Text))

            ' Calculate total leave taken
```

```
        Dim totalLeaveTaken As Integer = dutyDay + earlyLeave + casualLeave + holiday

        ' Calculate unpaid leave
        Dim unpaidLeave As Integer = deductedValue - totalLeaveTaken

        ' Display unpaid leave
        Label45.Text = unpaidLeave.ToString()
    Catch ex As Exception
        MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub

Private Sub TextBox_TextChanged(sender As Object, e As EventArgs) Handles TextBox3.TextChanged, TextBox4.TextChanged, TextBox5.TextChanged, TextBox6.TextChanged
    UpdateUnpaidLeave()
End Sub

Private Sub DisplayPFAndESIRates()
    Dim connString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    ' Create a connection and command object
    Dim connection As New SqlConnection(connString)

    Try
        ' Open connection
        connection.Open()

        ' Query to fetch PF rate from parameter table
        Dim queryPF As String = "SELECT [PF EMPLOYEE RATE] FROM Parameter"
        Dim commandPF As SqlCommand = New SqlCommand(queryPF, connection)
```

```
        Dim pfRate As Decimal = Convert.ToDecimal(commandPF.ExecuteScalar()) / 100 ' Divide by
100

        Label33.Text = pfRate.ToString()


        ' Query to fetch ESI rate from parameter table

        Dim queryESI As String = "SELECT [ESI EMPLOYEE RATE] FROM Parameter"

        Dim commandESI As SqlCommand = New SqlCommand(queryESI, connection)

        Dim esiRate As Decimal = Convert.ToDecimal(commandESI.ExecuteScalar()) / 100 ' Divide
by 100

        Label34.Text = esiRate.ToString()


    Catch ex As Exception

        MessageBox.Show("Error fetching rates from database: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)

    Finally

        ' Close connection

        connection.Close()

    End Try

  End Sub

  Private Sub UpdatePF()

    Dim connString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial
Catalog=PAYROLL;Integrated Security=True;Encrypt=False"




    ' Create a connection and command object

    Dim connection As New SqlConnection(connString)

    Try

        connection.Open()

        ' Convert Label8 and Label9 values to decimals

        Dim label8Value As Decimal = Decimal.Parse(Label8.Text)

        Dim label9Value As Decimal = Decimal.Parse(Label9.Text)


        ' Check if Label8 value is greater than 21000

        If label8Value > 21000 Then
```

```
            ' Query to fetch PF rate from parameter table
            Dim queryPF As String = "SELECT [PF EMPLOYEE RATE]  FROM Parameter"
            Dim commandPF As SqlCommand = New SqlCommand(queryPF, connection)
            Dim pfRate As Decimal = Convert.ToDecimal(commandPF.ExecuteScalar()) / 100 ' Divide by 100

            ' Calculate PF amount
            Dim pfAmount As Decimal = (label8Value + label9Value) * pfRate

            ' Display PF amount in Label31
            Label31.Text = pfAmount.ToString()
        Else
            ' If Label8 value is not greater than 21000, set Label31 value to zero
            Label31.Text = "0"
        End If
    Catch ex As Exception
        MessageBox.Show("Error calculating PF: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
Private Sub UpdateESI()
    Dim connString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
    Dim connection As New SqlConnection(connString)
    Try
        connection.Open()
        Dim label8Value As Decimal = Decimal.Parse(Label8.Text)
        Dim label9Value As Decimal = Decimal.Parse(Label9.Text)

        ' Query to fetch ESI rate from parameter table
        Dim queryESI As String = "SELECT [ESI EMPLOYEE RATE] FROM Parameter"
        Dim commandESI As SqlCommand = New SqlCommand(queryESI, connection)
        Dim esiRate As Decimal = Convert.ToDecimal(commandESI.ExecuteScalar()) / 100 ' Divide by 100
```

```
        ' Calculate ESI amount
        Dim esiAmount As Decimal = (label8Value + label9Value) * esiRate

        ' Display ESI amount in Label32
        Label32.Text = esiAmount.ToString()
    Catch ex As Exception
        MessageBox.Show("Error calculating ESI: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
Private Sub UpdateTotalDaysWorked()
    Try
        ' Convert TextBox inputs to integers
        Dim textBox1Value As Integer = If(String.IsNullOrEmpty(TextBox3.Text), 0, Integer.Parse(TextBox3.Text))
        Dim textBox2Value As Integer = If(String.IsNullOrEmpty(TextBox4.Text), 0, Integer.Parse(TextBox4.Text))
        Dim textBox3Value As Integer = If(String.IsNullOrEmpty(TextBox5.Text), 0, Integer.Parse(TextBox5.Text))
        Dim textBox4Value As Integer = If(String.IsNullOrEmpty(TextBox6.Text), 0, Integer.Parse(TextBox6.Text))

        ' Convert Label45 value to integer
        Dim label45Value As Integer = If(String.IsNullOrEmpty(Label45.Text), 0, Integer.Parse(Label45.Text))

        ' Calculate total days worked
        Dim totalDaysWorked As Integer = (textBox1Value + textBox2Value + textBox3Value + textBox4Value)

        ' Display total days worked in LabelXX (replace XX with the appropriate label number)

        ' Calculate salary
        Dim salary As Decimal = 0
        Dim labelValuesSum As Decimal = Decimal.Parse(Label8.Text) + Decimal.Parse(Label9.Text) + Decimal.Parse(Label10.Text) +
```

```
                    Decimal.Parse(Label11.Text) + Decimal.Parse(Label12.Text) +
Decimal.Parse(Label13.Text)

        Dim label2Value As Decimal = If(String.IsNullOrEmpty(Label41.Text), 0,
Decimal.Parse(Label41.Text))


        If label2Value <> 0 Then

           salary = (labelValuesSum / label2Value) * totalDaysWorked


        End If


        ' Display salary in LabelYY (replace YY with the appropriate label number)


        Dim textBox7Value As Integer = If(String.IsNullOrEmpty(TextBox8.Text), 0,
Integer.Parse(TextBox8.Text))

        Dim textBox8Value As Integer = If(String.IsNullOrEmpty(TextBox9.Text), 0,
Integer.Parse(TextBox9.Text))

        Dim textBox5Value As Integer = If(String.IsNullOrEmpty(TextBox10.Text), 0,
Integer.Parse(TextBox10.Text))

        Dim textBox6Value As Integer = If(String.IsNullOrEmpty(TextBox11.Text), 0,
Integer.Parse(TextBox11.Text))


        ' Convert Label45 value to integer

        Dim label8Value As Integer = If(String.IsNullOrEmpty(Label31.Text), 0,
Decimal.Parse(Label31.Text))

        Dim label9Value As Integer = If(String.IsNullOrEmpty(Label32.Text), 0,
Decimal.Parse(Label32.Text))

        Dim totalDEDUCTION As Integer = (textBox7Value + textBox8Value + textBox5Value +
textBox6Value + label8Value + label9Value)


        Dim TOTALSALARY As Integer = salary - totalDEDUCTION


        Label35.Text = TOTALSALARY.ToString()

    Catch ex As Exception

        MessageBox.Show("Error calculating total days worked and salary: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)

    End Try

End Sub
```

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Label35.Visible = True
    UpdateTotalDaysWorked()
    AddOrUpdateTransaction()
End Sub
Private Sub AddOrUpdateTransaction()
    Dim connString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
    Dim connection As New SqlConnection(connString)
    Try

        ' Convert inputs to appropriate data types
        Dim employeeCode As Integer
        If Integer.TryParse(TextBox2.Text, employeeCode) = False Then employeeCode = 0

        Dim month As String = If(ComboBox1.SelectedItem IsNot Nothing, ComboBox1.SelectedItem.ToString(), String.Empty)

        Dim year As Integer
        If Integer.TryParse(TextBox1.Text, year) = False Then year = 0

        Dim workingDay As Integer
        If Integer.TryParse(TextBox3.Text, workingDay) = False Then workingDay = 0

        Dim holiday As Integer
        If Integer.TryParse(TextBox4.Text, holiday) = False Then holiday = 0

        Dim earlyLeave As Integer
        If Integer.TryParse(TextBox5.Text, earlyLeave) = False Then earlyLeave = 0

        Dim casualLeave As Integer
        If Integer.TryParse(TextBox6.Text, casualLeave) = False Then casualLeave = 0

        Dim leaveWithoutPay As Integer
```

```
        If Integer.TryParse(Label45.Text, leaveWithoutPay) = False Then leaveWithoutPay = 0

        Dim tds As Decimal
        If Decimal.TryParse(TextBox8.Text, tds) = False Then tds = 0

        Dim advance As Decimal
        If Decimal.TryParse(TextBox9.Text, advance) = False Then advance = 0

        Dim loanFromBank As Decimal
        If Decimal.TryParse(TextBox10.Text, loanFromBank) = False Then loanFromBank = 0

        Dim loanFromCompany As Decimal
        If Decimal.TryParse(TextBox11.Text, loanFromCompany) = False Then loanFromCompany =
0

        Dim totalSalary As Decimal
        If Decimal.TryParse(Label35.Text, totalSalary) = False Then totalSalary = 0

        ' Open the connection
        connection.Open()

        ' Check if record exists
        Dim queryCheck As String = "SELECT COUNT(*) FROM [TRANSACTION] WHERE
Employee_Code = @Employee_Code AND [Month] = @Month AND [Year] = @Year"
        Dim commandCheck As SqlCommand = New SqlCommand(queryCheck, connection)
        commandCheck.Parameters.AddWithValue("@Employee_Code", employeeCode)
        commandCheck.Parameters.AddWithValue("@Month", month)
        commandCheck.Parameters.AddWithValue("@Year", year)

        Dim recordCount As Integer = Convert.ToInt32(commandCheck.ExecuteScalar())

        If recordCount > 0 Then
            ' Record exists, ask for update
```

```
        Dim dialogResult As DialogResult = MessageBox.Show("Record for this Employee Code, Month, and Year already exists. Do you want to update it?", "Update Record", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

        If dialogResult = DialogResult.Yes Then

            ' Update existing record

            Dim queryUpdate As String = "UPDATE [Transaction] SET Working_Day = @Working_Day, Holiday = @Holiday, Early_Leave = @Early_Leave, Casual_Leave = @Casual_Leave, Leave_Without_Pay = @Leave_Without_Pay, TDS = @TDS, Advance = @Advance, Loan_From_Bank = @Loan_From_Bank, Loan_From_Company = @Loan_From_Company, Total_Salary = @Total_Salary WHERE Employee_Code = @Employee_Code AND [Month] = @Month AND [Year] = @Year"

            Dim commandUpdate As SqlCommand = New SqlCommand(queryUpdate, connection)

            commandUpdate.Parameters.AddWithValue("@Working_Day", workingDay)

            commandUpdate.Parameters.AddWithValue("@Holiday", holiday)

            commandUpdate.Parameters.AddWithValue("@Early_Leave", earlyLeave)

            commandUpdate.Parameters.AddWithValue("@Casual_Leave", casualLeave)

            commandUpdate.Parameters.AddWithValue("@Leave_Without_Pay", leaveWithoutPay)

            commandUpdate.Parameters.AddWithValue("@TDS", tds)

            commandUpdate.Parameters.AddWithValue("@Advance", advance)

            commandUpdate.Parameters.AddWithValue("@Loan_From_Bank", loanFromBank)

            commandUpdate.Parameters.AddWithValue("@Loan_From_Company", loanFromCompany)

            commandUpdate.Parameters.AddWithValue("@Total_Salary", totalSalary)

            commandUpdate.Parameters.AddWithValue("@Employee_Code", employeeCode)

            commandUpdate.Parameters.AddWithValue("@Month", month)

            commandUpdate.Parameters.AddWithValue("@Year", year)


            commandUpdate.ExecuteNonQuery()


            MessageBox.Show("Record updated successfully.", "Update Success", MessageBoxButtons.OK, MessageBoxIcon.Information)

        End If

    Else

        ' Insert new record

        Dim queryInsert As String = "INSERT INTO [Transaction] (Employee_Code, [Month], [Year], Working_Day, Holiday, Early_Leave, Casual_Leave, Leave_Without_Pay, TDS, Advance, Loan_From_Bank, Loan_From_Company, Total_Salary) VALUES (@Employee_Code, @Month, @Year, @Working_Day, @Holiday, @Early_Leave, @Casual_Leave, @Leave_Without_Pay, @TDS, @Advance, @Loan_From_Bank, @Loan_From_Company, @Total_Salary)"
```

```
            Dim commandInsert As SqlCommand = New SqlCommand(queryInsert, connection)
            commandInsert.Parameters.AddWithValue("@Employee_Code", employeeCode)
            commandInsert.Parameters.AddWithValue("@Month", month)
            commandInsert.Parameters.AddWithValue("@Year", year)
            commandInsert.Parameters.AddWithValue("@Working_Day", workingDay)
            commandInsert.Parameters.AddWithValue("@Holiday", holiday)
            commandInsert.Parameters.AddWithValue("@Early_Leave", earlyLeave)
            commandInsert.Parameters.AddWithValue("@Casual_Leave", casualLeave)
            commandInsert.Parameters.AddWithValue("@Leave_Without_Pay", leaveWithoutPay)
            commandInsert.Parameters.AddWithValue("@TDS", tds)
            commandInsert.Parameters.AddWithValue("@Advance", advance)
            commandInsert.Parameters.AddWithValue("@Loan_From_Bank", loanFromBank)
            commandInsert.Parameters.AddWithValue("@Loan_From_Company",
loanFromCompany)
            commandInsert.Parameters.AddWithValue("@Total_Salary", totalSalary)


            commandInsert.ExecuteNonQuery()


            MessageBox.Show("Record added successfully.", "Insert Success",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If


    Finally
        ' Close the connection
        connection.Close()
    End Try
  End Sub


End Class
```

12-REPORT:-

```
Imports System.Data.SqlClient

Public Class REPORT
    ' Replace "connectionstring" with your actual connection string
    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"

    Private Sub REPORT_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadCompanyDetails()
        ComboBox1.Items.AddRange(New String() {"January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"})

        ' Set the default selected item to the current month
        Dim currentMonth As Integer = DateTime.Now.Month
        ComboBox1.SelectedIndex = currentMonth - 1

        ' Set the default value of the TextBox to the current year
        TextBox1.Text = DateTime.Now.Year.ToString()

        ' Initially hide DataGridViews and Labels
        DataGridView1.Visible = False
        DataGridView2.Visible = False
        DataGridView3.Visible = False
        DataGridView4.Visible = False
        DataGridView5.Visible = False
        Label3.Visible = False
        Label4.Visible = False
        Label5.Visible = False
        Label6.Visible = False
        Label7.Visible = False

        ' Initialize empty DataTables for each department
        DataGridView1.DataSource = dtHR
        DataGridView2.DataSource = dtProduction
        DataGridView3.DataSource = dtIT
        DataGridView4.DataSource = dtFinance
        DataGridView5.DataSource = dtSales
    End Sub

    Dim dtHR As New DataTable("HR")
    Dim dtProduction As New DataTable("Production")
    Dim dtIT As New DataTable("IT")
    Dim dtFinance As New DataTable("Finance")
    Dim dtSales As New DataTable("Sales")

    Sub LoadData()
        Dim selectedMonth As String = ComboBox1.SelectedItem.ToString()
        Dim selectedYear As Integer = Integer.Parse(TextBox1.Text)

        ' Make DataGridViews and Labels visible
        DataGridView1.Visible = True
        DataGridView2.Visible = True
        DataGridView3.Visible = True
```

```
        DataGridView4.Visible = True
        DataGridView5.Visible = True
        Label3.Visible = True
        Label4.Visible = True
        Label5.Visible = True
        Label6.Visible = True
        Label7.Visible = True

        Dim connection As SqlConnection = New SqlConnection(connectionString)

        ' Build the base SQL statement
        Dim sql = "SELECT e.EMPLOYEE_CODE, e.EMPLOYEE_NAME, e.Department, " &
          " s.[DESIGNATION], s.[BASIC_SALARY], s.[DA], " &
          "s.[HRA], s.[MEDICAL], s.[PERFORMANCE], s.[CONVEYANCE], " &
          "t.MONTH, t.YEAR, t.WORKING_DAY, t.HOLIDAY, t.EARLY_LEAVE,
t.CASUAL_LEAVE, " &
          "t.LEAVE_WITHOUT_PAY, t.TDS, t.ADVANCE, t.LOAN_FROM_BANK,
t.LOAN_FROM_COMPANY, " &
          "t.TOTAL_SALARY " &
          "FROM EMPLOYEE1 e " &
          "INNER JOIN SALARY s ON e.Employee_CODE = s.Employee_CODE " &
          "INNER JOIN [TRANSACTION] t ON e.Employee_CODE = t.Employee_CODE " &
          "WHERE t.MONTH = @Month AND t.YEAR = @Year " &
          "AND e.Department = @Department " &
          "ORDER BY e.EMPLOYEE_NAME"


        ' Create separate DataAdapters for each department
        Dim adapterHR As New SqlDataAdapter(sql, connection)
        Dim adapterProduction As New SqlDataAdapter(sql, connection)
        Dim adapterIT As New SqlDataAdapter(sql, connection)
        Dim adapterFinance As New SqlDataAdapter(sql, connection)
        Dim adapterSales As New SqlDataAdapter(sql, connection)

        ' Add parameters for month and year to each adapter
        adapterHR.SelectCommand.Parameters.AddWithValue("@Month", selectedMonth)
        adapterHR.SelectCommand.Parameters.AddWithValue("@Year", selectedYear)
        adapterHR.SelectCommand.Parameters.AddWithValue("@Department",
"HUMAN RESOURCE")

        adapterProduction.SelectCommand.Parameters.AddWithValue("@Month", selectedMonth)
        adapterProduction.SelectCommand.Parameters.AddWithValue("@Year", selectedYear)
        adapterProduction.SelectCommand.Parameters.AddWithValue("@Department",
"PRODUCTION")

        adapterIT.SelectCommand.Parameters.AddWithValue("@Month", selectedMonth)
        adapterIT.SelectCommand.Parameters.AddWithValue("@Year", selectedYear)
        adapterIT.SelectCommand.Parameters.AddWithValue("@Department", "INFORMATION AND
TECHNOLOGY")

        adapterFinance.SelectCommand.Parameters.AddWithValue("@Month", selectedMonth)
        adapterFinance.SelectCommand.Parameters.AddWithValue("@Year", selectedYear)
        adapterFinance.SelectCommand.Parameters.AddWithValue("@Department", "FINANCE")

        adapterSales.SelectCommand.Parameters.AddWithValue("@Month", selectedMonth)
```

```
        adapterSales.SelectCommand.Parameters.AddWithValue("@Year", selectedYear)
        adapterSales.SelectCommand.Parameters.AddWithValue("@Department", "SALES")

        ' Clear previous data from DataTables
        dtHR.Clear()
        dtProduction.Clear()
        dtIT.Clear()
        dtFinance.Clear()
        dtSales.Clear()

        ' Fill each DataTable with data from its corresponding adapter
        adapterHR.Fill(dtHR)
        adapterProduction.Fill(dtProduction)
        adapterIT.Fill(dtIT)
        adapterFinance.Fill(dtFinance)
        adapterSales.Fill(dtSales)
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        LoadData()
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Me.Close()
        HOME.Show()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ' Get selected month
        Dim selectedMonth As String = If(ComboBox1.SelectedItem IsNot Nothing,
ComboBox1.SelectedItem.ToString(), "")
        If String.IsNullOrEmpty(selectedMonth) Then
            MessageBox.Show("Please select a month.")
            Return ' Exit the function if no month is selected
        End If

        ' Convert year (assuming numeric)
        Dim selectedYear As Integer
        If Not Integer.TryParse(TextBox1.Text, selectedYear) Then
            MessageBox.Show("Invalid year format. Please enter a valid number.")
            Return ' Exit the function if conversion fails
        End If

        ' Pass month and year to FINAL form
        Dim finalForm As New FINAL(selectedMonth, selectedYear)
        finalForm.Show()
        Me.Close()
    End Sub
    Private Sub LoadCompanyDetails()
        ' SQL query to get the company name and address
        Dim query As String = "SELECT [COMPANY NAME], [ADDRESS] FROM
[dbo].[COMPANY]"

        ' Create a connection and command object
        Using connection As New SqlConnection(connectionString),
```

```
            command As New SqlCommand(query, connection)
        Try
            ' Open the connection
            connection.Open()

            ' Execute the command and get the data reader
            Using reader As SqlDataReader = command.ExecuteReader()
                ' Check if there is a row to read
                If reader.Read() Then
                    ' Get the name and address from the reader
                    Dim companyName As String = reader("COMPANY NAME").ToString()
                    Dim companyAddress As String = reader("ADDRESS").ToString()

                    ' Set the labels' text to the company details
                    Label8.Text = companyName
                    Label9.Text = companyAddress
                Else
                    ' Handle the case where no data is returned
                    MessageBox.Show("No company details found.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                End If
            End Using

        Catch ex As Exception
            ' Handle any errors that might have occurred
            MessageBox.Show("An error occurred while fetching company details: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Using
  End Sub
End Class
```

13-SUMMARY:-

```
Imports System.Windows.Forms
Imports System.Data.SqlClient


Public Class FINAL
    Private _selectedMonth As String
    Private _selectedYear As Integer
    Private connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
    Public Sub New(selectedMonth As String, selectedYear As Integer)
        InitializeComponent()
        _selectedMonth = selectedMonth
        _selectedYear = selectedYear
        Label18.Text = _selectedMonth
        Label19.Text = _selectedYear.ToString()  ' Ensure conversion to string for display
    End Sub


    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        REPORT.Show()
    End Sub


    Private Sub FINAL_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Label18.Visible = False
        Label19.Visible = False
        LoadCompanyDetails()
        Dim selectedMonth As String = Label18.Text
        Dim selectedYear As Integer = 0
        If Not Integer.TryParse(Label19.Text, selectedYear) Then
            MessageBox.Show("Invalid year format. Please enter a valid number.")
            Return  ' Exit the function if conversion fails
        End If
        Dim hrData As Tuple(Of Integer, Integer) = GetHREmployeeCount(selectedMonth, selectedYear)
```

```
        ' Check for data retrieval success (optional)
        If hrData Is Nothing Then
            MessageBox.Show("Error retrieving data")
        Else
            Label6.Text = hrData.Item1.ToString()  ' Display HR employee count
            Label15.Text = hrData.Item2.ToString()  ' Display total working days
        End If
        Dim PRData As Tuple(Of Integer, Integer) = GetPRODUCTIONEmployeeCount(selectedMonth, selectedYear)

        ' Check for data retrieval success (optional)
        If PRData Is Nothing Then
            MessageBox.Show("Error retrieving data")
        Else
            Label7.Text = PRData.Item1.ToString()  ' Display HR employee count
            Label14.Text = PRData.Item2.ToString()  ' Display total working days
        End If
        Dim SLData As Tuple(Of Integer, Integer) = GetSALESEmployeeCount(selectedMonth, selectedYear)

        ' Check for data retrieval success (optional)
        If SLData Is Nothing Then
            MessageBox.Show("Error retrieving data")
        Else
            Label8.Text = SLData.Item1.ToString()  ' Display HR employee count
            Label13.Text = SLData.Item2.ToString()  ' Display total working days
        End If
        Dim FIData As Tuple(Of Integer, Integer) = GetFINANCEEmployeeCount(selectedMonth, selectedYear)

        ' Check for data retrieval success (optional)
        If FIData Is Nothing Then
            MessageBox.Show("Error retrieving data")
        Else
```

```
        Label9.Text = FIData.Item1.ToString()  ' Display HR employee count
        Label12.Text = FIData.Item2.ToString()  ' Display total working days
    End If
    Dim ITData As Tuple(Of Integer, Integer) = GetITEmployeeCount(selectedMonth, selectedYear)
    Dim sum As Integer = 0 ' Initialize sum variable

    ' Check for data retrieval success (optional)
    If ITData Is Nothing Then
        MessageBox.Show("Error retrieving data")
    Else
        Label10.Text = ITData.Item1.ToString()  ' Display HR employee count
        Label11.Text = ITData.Item2.ToString()  ' Display total working days

        ' Convert label texts to integers and sum them up
        sum = Convert.ToInt32(Label15.Text) + Convert.ToInt32(Label14.Text) + Convert.ToInt32(Label13.Text) + Convert.ToInt32(Label12.Text) + Convert.ToInt32(Label11.Text)
    End If

    Label16.Text = sum.ToString()

End Sub

Public Function GetHREmployeeCount(ByVal selectedMonth As String, ByVal selectedYear As Integer) As Tuple(Of Integer, Integer)
    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"  ' Update with your actual connection string

    ' SQL query to count HR employees and total working days, filtered by month and year
    Dim sql As String = "Select COUNT(e.Employee_CODE) As EmployeeCount, " &
"SUM(T.TOTAL_SALARY) As TotalSalary " &
"FROM EMPLOYEE1 e " &
"INNER JOIN SALARY s On e.Employee_CODE = s.Employee_CODE " &
"INNER JOIN [TRANSACTION] t On e.Employee_CODE = t.Employee_CODE " &
```

```
"WHERE t.MONTH = @Month And t.YEAR = @Year And e.Department = 'HUMAN RESOURCE'"


        ' Execute the query and retrieve data
        Using connection As New SqlConnection(connectionString)
            Using command As New SqlCommand(sql, connection)
                command.Parameters.AddWithValue("@Month", selectedMonth)
                command.Parameters.AddWithValue("@Year", selectedYear)
                connection.Open()
                Dim reader As SqlDataReader = command.ExecuteReader()


                If reader.Read() Then
                    Dim employeeCount As Integer = If(IsDBNull(reader("EmployeeCount")), 0, Convert.ToInt32(reader("EmployeeCount")))
                    Dim TotalSalary As Integer = If(IsDBNull(reader("TotalSalary")), 0, Convert.ToInt32(reader("TotalSalary")))
                    Return New Tuple(Of Integer, Integer)(employeeCount, TotalSalary)
                Else
                    ' Handle case where no data is found (optional)
                    Return New Tuple(Of Integer, Integer)(0, 0)  ' Return 0 for both counts
                End If


                reader.Close()
            End Using
        End Using


        ' Should not reach here if data is retrieved successfully
        Return Nothing  ' Indicate an error (optional)
    End Function
    Public Function GetPRODUCTIONEmployeeCount(ByVal selectedMonth As String, ByVal selectedYear As Integer) As Tuple(Of Integer, Integer)
        Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"  ' Update with your actual connection string


        ' SQL query to count HR employees and total working days, filtered by month and year
```

```
    Dim sql As String = "SELECT COUNT(e.Employee_CODE) AS EmployeeCount, " &
"SUM(T.TOTAL_SALARY) AS TotalSalary " &
"FROM EMPLOYEE1 e " &
"INNER JOIN SALARY s ON e.Employee_CODE = s.Employee_CODE " &
"INNER JOIN [TRANSACTION] t ON e.Employee_CODE = t.Employee_CODE " &
"WHERE t.MONTH = @Month AND t.YEAR = @Year AND e.Department = 'PRODUCTION'"


    ' Execute the query and retrieve data
    Using connection As New SqlConnection(connectionString)
        Using command As New SqlCommand(sql, connection)
            command.Parameters.AddWithValue("@Month", selectedMonth)
            command.Parameters.AddWithValue("@Year", selectedYear)
            connection.Open()
            Dim reader As SqlDataReader = command.ExecuteReader()

            If reader.Read() Then
                Dim employeeCount As Integer = If(IsDBNull(reader("EmployeeCount")), 0,
Convert.ToInt32(reader("EmployeeCount")))
                Dim TotalSalary As Integer = If(IsDBNull(reader("TotalSalary")), 0,
Convert.ToInt32(reader("TotalSalary")))
                Return New Tuple(Of Integer, Integer)(employeeCount, TotalSalary)
            Else
                ' Handle case where no data is found (optional)
                Return New Tuple(Of Integer, Integer)(0, 0)  ' Return 0 for both counts
            End If

            reader.Close()
        End Using
    End Using

    ' Should not reach here if data is retrieved successfully
    Return Nothing  ' Indicate an error (optional)
End Function
```

```
Public Function GetSALESEmployeeCount(ByVal selectedMonth As String, ByVal selectedYear As Integer) As Tuple(Of Integer, Integer)

    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"  ' Update with your actual connection string


    ' SQL query to count HR employees and total working days, filtered by month and year

    Dim sql As String = "SELECT COUNT(e.Employee_CODE) AS EmployeeCount, " &

"SUM(T.TOTAL_SALARY) AS TotalSalary " &

"FROM EMPLOYEE1 e " &

"INNER JOIN SALARY s ON e.Employee_CODE = s.Employee_CODE " &

"INNER JOIN [TRANSACTION] t ON e.Employee_CODE = t.Employee_CODE " &

"WHERE t.MONTH = @Month AND t.YEAR = @Year AND e.Department = 'SALES'"



    ' Execute the query and retrieve data

    Using connection As New SqlConnection(connectionString)

        Using command As New SqlCommand(sql, connection)

            command.Parameters.AddWithValue("@Month", selectedMonth)

            command.Parameters.AddWithValue("@Year", selectedYear)

            connection.Open()

            Dim reader As SqlDataReader = command.ExecuteReader()


            If reader.Read() Then

                Dim employeeCount As Integer = If(IsDBNull(reader("EmployeeCount")), 0, Convert.ToInt32(reader("EmployeeCount")))

                Dim TotalSalary As Integer = If(IsDBNull(reader("TotalSalary")), 0, Convert.ToInt32(reader("TotalSalary")))

                Return New Tuple(Of Integer, Integer)(employeeCount, TotalSalary)

            Else

                ' Handle case where no data is found (optional)

                Return New Tuple(Of Integer, Integer)(0, 0)  ' Return 0 for both counts

            End If


            reader.Close()

        End Using
```

```
        Return Nothing
    End Using ' Indicate an error (optional)
End Function

Public Function GetFINANCEEmployeeCount(ByVal selectedMonth As String, ByVal selectedYear As Integer) As Tuple(Of Integer, Integer)
    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"  ' Update with your actual connection string

    ' SQL query to count HR employees and total working days, filtered by month and year
    Dim sql As String = "SELECT COUNT(e.Employee_CODE) AS EmployeeCount, " &
"SUM(T.TOTAL_SALARY) AS TotalSalary " &
"FROM EMPLOYEE1 e " &
"INNER JOIN SALARY s ON e.Employee_CODE = s.Employee_CODE " &
"INNER JOIN [TRANSACTION] t ON e.Employee_CODE = t.Employee_CODE " &
"WHERE t.MONTH = @Month AND t.YEAR = @Year AND e.Department = 'FINANCE'"

    ' Execute the query and retrieve data
    Using connection As New SqlConnection(connectionString)
        Using command As New SqlCommand(sql, connection)
            command.Parameters.AddWithValue("@Month", selectedMonth)
            command.Parameters.AddWithValue("@Year", selectedYear)
            connection.Open()
            Dim reader As SqlDataReader = command.ExecuteReader()

            If reader.Read() Then
                Dim employeeCount As Integer = If(IsDBNull(reader("EmployeeCount")), 0, Convert.ToInt32(reader("EmployeeCount")))
                Dim TotalSalary As Integer = If(IsDBNull(reader("TotalSalary")), 0, Convert.ToInt32(reader("TotalSalary")))
                Return New Tuple(Of Integer, Integer)(employeeCount, TotalSalary)
            Else
                ' Handle case where no data is found (optional)
                Return New Tuple(Of Integer, Integer)(0, 0)  ' Return 0 for both counts
```

```
            End If

            reader.Close()
        End Using
    End Using

    ' Should not reach here if data is retrieved successfully

    ' Should not reach here if data is retrieved successfully
    Return Nothing  ' Indicate an error (optional)
End Function

Public Function GetITEmployeeCount(ByVal selectedMonth As String, ByVal selectedYear As Integer) As Tuple(Of Integer, Integer)
    Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"  ' Update with your actual connection string

    ' SQL query to count HR employees and total working days, filtered by month and year
    Dim sql As String = "SELECT COUNT(e.Employee_CODE) AS EmployeeCount, " &
"SUM(T.TOTAL_SALARY) AS TotalSalary " &
"FROM EMPLOYEE1 e " &
"INNER JOIN SALARY s ON e.Employee_CODE = s.Employee_CODE " &
"INNER JOIN [TRANSACTION] t ON e.Employee_CODE = t.Employee_CODE " &
"WHERE t.MONTH = @Month AND t.YEAR = @Year AND e.Department = 'INFORMATION AND TECHNOLOGY'"

    ' Execute the query and retrieve data
    Using connection As New SqlConnection(connectionString)
        Using command As New SqlCommand(sql, connection)
            command.Parameters.AddWithValue("@Month", selectedMonth)
            command.Parameters.AddWithValue("@Year", selectedYear)
            connection.Open()
            Dim reader As SqlDataReader = command.ExecuteReader()

            If reader.Read() Then
```

```
            Dim employeeCount As Integer = If(IsDBNull(reader("EmployeeCount")), 0, Convert.ToInt32(reader("EmployeeCount")))
            Dim TotalSalary As Integer = If(IsDBNull(reader("TotalSalary")), 0, Convert.ToInt32(reader("TotalSalary")))
            Return New Tuple(Of Integer, Integer)(employeeCount, TotalSalary)
        Else
            ' Handle case where no data is found (optional)
            Return New Tuple(Of Integer, Integer)(0, 0)  ' Return 0 for both counts
        End If

        reader.Close()
      End Using
    End Using

    ' Should not reach here if data is retrieved successfully
    Return Nothing  ' Indicate an error (optional)
  End Function
  Private Sub LoadCompanyDetails()
    ' SQL query to get the company name and address
    Dim query As String = "SELECT [COMPANY NAME], [ADDRESS] FROM [dbo].[COMPANY]"

    ' Create a connection and command object
    Using connection As New SqlConnection(connectionString),
        command As New SqlCommand(query, connection)
      Try
        ' Open the connection
        connection.Open()

        ' Execute the command and get the data reader
        Using reader As SqlDataReader = command.ExecuteReader()
          ' Check if there is a row to read
          If reader.Read() Then
            ' Get the name and address from the reader
            Dim companyName As String = reader("COMPANY NAME").ToString()
```

```
                Dim companyAddress As String = reader("ADDRESS").ToString()


                ' Set the labels' text to the company details
                Label23.Text = companyName
                Label24.Text = companyAddress
            Else
                ' Handle the case where no data is returned
                MessageBox.Show("No company details found.", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information)
            End If
        End Using


    Catch ex As Exception
        ' Handle any errors that might have occurred
        MessageBox.Show("An error occurred while fetching company details: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
  End Using
End Sub


End Class
```

14-LEARN:-

```
Imports System.Data.SqlClient

Public Class LEARN
   Dim connectionString As String = "Data Source=DESKTOP-DL4DPHJ\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True;Encrypt=False"
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
      Me.Close()
      HOME.Show()
   End Sub

   Private Sub LEARN_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      LoadCompanyDetails()
   End Sub
   Private Sub LoadCompanyDetails()
      ' SQL query to get the company name and address
      Dim query As String = "SELECT [COMPANY NAME], [PHONE] FROM [dbo].[COMPANY]"

      ' Create a connection and command object
      Using connection As New SqlConnection(connectionString),
          command As New SqlCommand(query, connection)
         Try
            ' Open the connection
            connection.Open()

            ' Execute the command and get the data reader
            Using reader As SqlDataReader = command.ExecuteReader()
               ' Check if there is a row to read
               If reader.Read() Then
                  ' Get the name and address from the reader
                  Dim companyName As String = reader("COMPANY NAME").ToString()
                  Dim companyPHONE As String = reader("PHONE").ToString()

                  ' Set the labels' text to the company details
                  Label18.Text = companyName
                  Label17.Text = companyPHONE
               Else
                  ' Handle the case where no data is returned
                  MessageBox.Show("No company details found.", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information)
               End If
            End Using

         Catch ex As Exception
            ' Handle any errors that might have occurred
            MessageBox.Show("An error occurred while fetching company details: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
         End Try
      End Using
   End Sub
End Class
```

CONCLUSION

Reflecting on my internship at Motilal Dulichand, developing the payroll management system was a pivotal learning experience in software development and database integration. Although it was a training project, it provided me with essential skills in creating intuitive interfaces, handling data efficiently, and collaborating effectively within a corporate environment.

The system aimed to automate the management of employee details, streamline salary calculations, and generate insightful reports for HR administrators. Using Visual Basic for frontend development, I focused on designing a user-friendly interface that facilitated seamless interaction. On the backend, SQL ensured secure and efficient data storage and retrieval, crucial for maintaining accurate employee records and salary information.

Throughout the project, I worked closely with HR professionals to align the system with industry standards and organizational needs. This collaboration not only enhanced the system's functionality but also emphasized the importance of user feedback in refining software solutions. Simulating real-world payroll scenarios allowed me to tackle challenges in software lifecycle management, reinforcing my understanding of project execution and problem-solving in IT.

In conclusion, while the payroll management system served primarily as a learning tool, it laid a solid foundation for my career in software development. It deepened my technical skills, broadened my perspective on technology's role in business optimization, and fueled my enthusiasm to contribute meaningfully to future IT projects.