

PAYROLL MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by:

- AKANSHA KUMARI : 18BCA001A

*Submitted in partial fulfilment of the requirements for the
Award of the degree of*

BACHELOR OF COMPUTER APPLICATION



*JYOTI NIVAS COLLEGE AUTONOMOUS
KORMANGALA, BANGALORE*

ACKNOWLEDGEMENT

With a deep sense of gratitude we acknowledge all those who have contributed significantly towards the successful completion of this project.

We would like to express our gratitude to **Dr. Sr. Elizabeth CS, Principal, Jyoti Nivas College** for the giving us the opportunity to complete our project successfully by providing us with endless enthusiastic support in improving our creative skills and for providing us the excellent facilities, and constant encouragement throughout our project.

We owe profound thanks to **Mrs. Neetha Georhin** the Head of the **Department of Computer Science**, who took keen interest on our project work and guided us and encouraged at every step all along till the completion of the project work by providing all needed information for developing a good system.

We express our heartfelt gratitude to our project guides, **Mrs. RUBY PETER AND MR.MARIA CHARLES DOMINIC** for their cordial and considerate attitude with all valuable suggestions, guidance support and cooperation right from the inception of the project, which inspired us in carrying out this project successfully within the short period of time.

Finally we are grateful to our parents and friends who have helped us to complete this project successfully.

INDEX

SL.NO	CONTENT	PAGE .NO
1.	SYNOPSIS	6
2.	SYSTEM SPECIFICATION a)hardware specification b)software specification c)network specification	8
3.	SYSTEM ANALYSIS a)Existing system b)Proposed system	9
4.	SYSTEM DESIGN a)database design b)ER diagram c>User Interface Designs d)code design	10-63
5.	SYSTEM TESTING	64
6.	FUTURE ENHANCEMENTS	67

SYNOPSIS

ABSTRACT

This document fully and formally describes the requirements of the proposed project system. Payroll management system is one of the core areas of business; it is pursued to manage employees' deduction, expenses, allowance, tax, salary etc, for a specific period of time. Management and accounting are the two main essential parts for payroll. Further this project will develop for company management and enhance business in market and to keep the reputation of the company.

PROJECT DEFINITION

A payroll system is software designed to organize all the tasks of employee payment and the filing of employee taxes.

It is basically payment of employees' by their employers. This task can include calculating wages, tracking the working hours, withhold tax and deduction of salary, delivering of checks and paying employment taxes to the government.

Payroll software often requires very little input from the employer. The employer is required to insert the required input of the employee wages information and hours then the software calculates the information and performs withholdings automatically most payroll software is automatically updated.

PROJECT DESCRIPTION

The payroll management system deals with the financial aspects of employees' salary, deductions, allowance, gross pay, bonus, net pay etc and generating payslip for certain duration or specified time period.

The benefit of payroll management system is its easy implementations other advantages of payroll management system are its extensive features and reports.

PURPOSE OF PAYROLL MANAGEMENT SYSTEM

Payroll management system gives you the power to:

- Manage employee information efficiently.
- Define the emulations, deductions, leave etc.
- Generate pay-slip at the convenience of a mouse click
- To Generate and manage the payroll processes accordingly, and the salary structure assigned to the employee.
- To generate all the reports related to employee, attendance or leave, payroll etc.
- Manage your own security

When deciding which system has to be chosen there are some factors to be considered.

First, analyse the size of your business and decide how much you are willing to spend on payroll processing, while it's possible for smaller business to handle payroll duties in-house through a manual process, much time can be wasted while attempting to calculate everything correctly.

One miscalculation and the business owner lead to law consequences. Mid-sized companies with up to 100 employees benefits greatly by investing in a payroll system

SYSTEM SPECIFICATION

- HARDWARE SPECIFICATION

- RAM :1GB
- HARD DISK: MINIMUM 20GB FREE SPACE
- PROCESSOR: PENTIUM4(1.6GHz)
- INTEL 8085 CHIP
- LOGITECH OPTICAL MOUSE
- KEYBOARD

- SOFTWARE SPECIFICATION

- OPERATING SYSTEM: WINDOWS XP
- FRONT END SOFTWARE: MICROSOFT VISUAL STUDIO
- BACK END SOFTWARE:SQL
- DATABASE: MICROSOFT SQL SERVER AND MYSQL

- NETWORK SPECIFICATION:

This software work stand-alone system.

SYSTEM ANALYSIS

Existing system

Enables your business to handle all your **employee's** financial records in a hassle-free, automated fashion. This includes **employee's** salaries, bonuses, deductions, net pay, and generation of pay-slips for a specific period.

Proposed System

It can propose in word document, computer science, management system, managing personal finance plan.

MODULE

The payroll module is capable of generating the payroll of the employee. The comprehensive module has link with sub features of the module.

Such as employee management, attendance detail management, payroll calculator, salary management, manager and the admin which helps to generates the salary payslips of an employee.

- It is important that further module to be used the basic data is available
- Modules are the key features of a payroll.

They are 5 modules in the payroll management system:

1. Employee management – employee management contains the records of individual employee.

- Processing all the records of the employee
- Generating the reports
- Generating the payslips
- Applying of leave

2. Attendance management – In this module the employee can view employee details.

- Employee can view the attendance.
- Modify the attendance is possible.
- Generating leave application is possible.

3. Department management-In this module the employee can view the department details.

- Generating the departmental reports
- Assigning the projects to the employee.

4. Salary management –in this module the salary of an employee is stored.

- Processing of allowance
- Processing of deduction based on leave

5. Payslip-A payslip is a note provided to an employee, with details such as the amount they've been paid for a particular period, as well as the amount of tax deducted.

- A payslip can be in either paper or digital format, and may be sent via email or post.
- Employee can print the payslips.

Data flow diagram

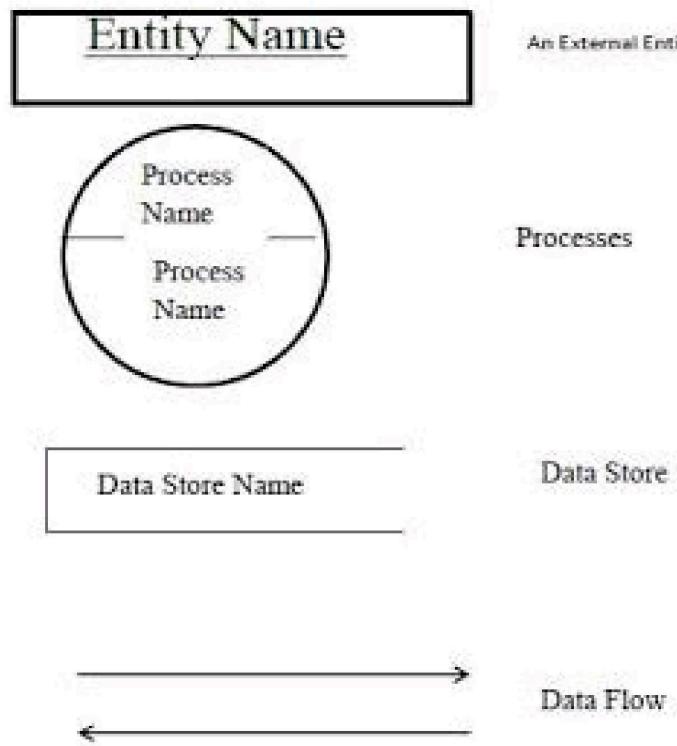
DFD provides a logical method of the system and shows the flow of data and the flow of logic involved.

Symbols used in dfd:

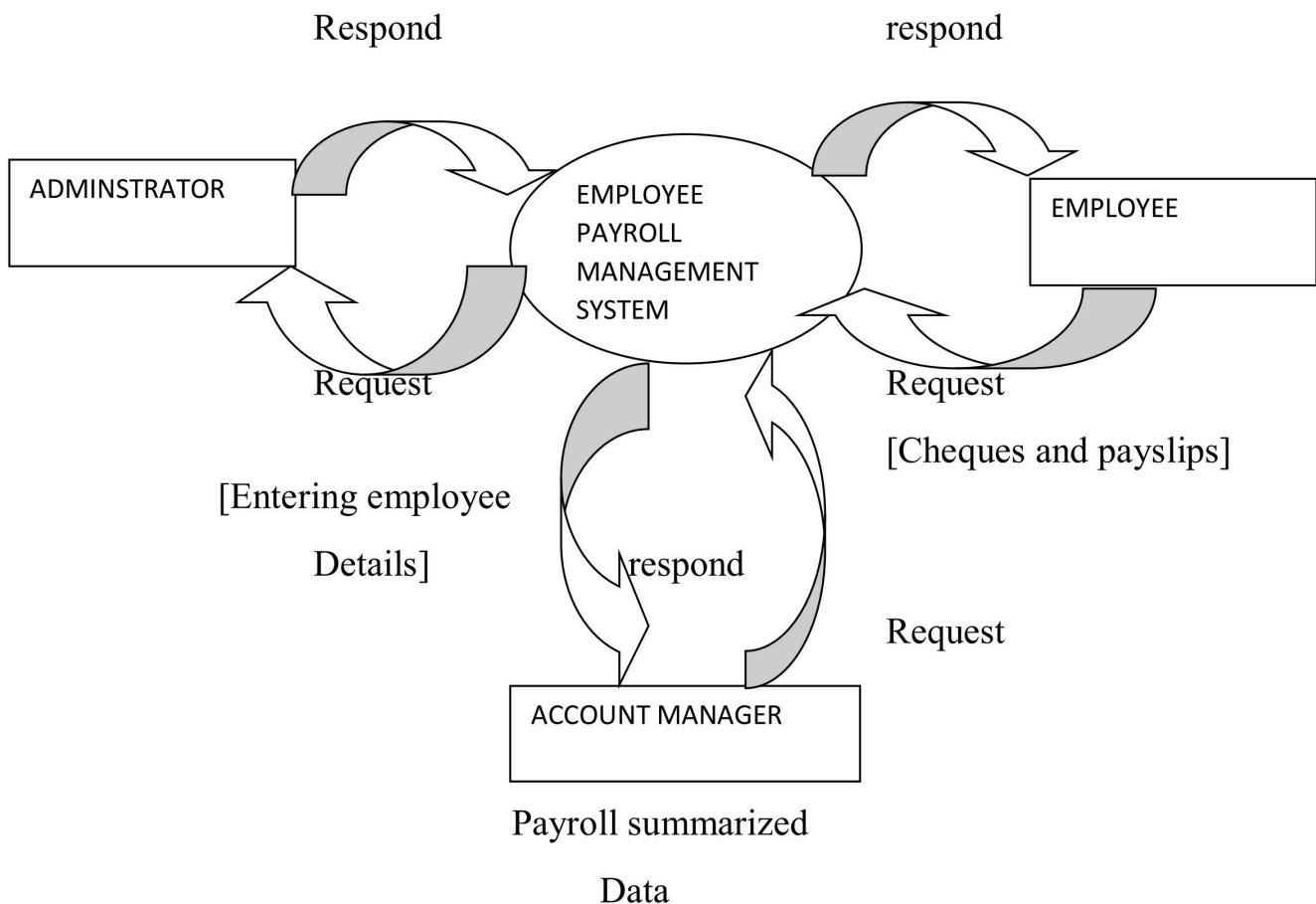
Four symbols are used in drawing dataflow diagrams, these are:

Symbols used in dfd:

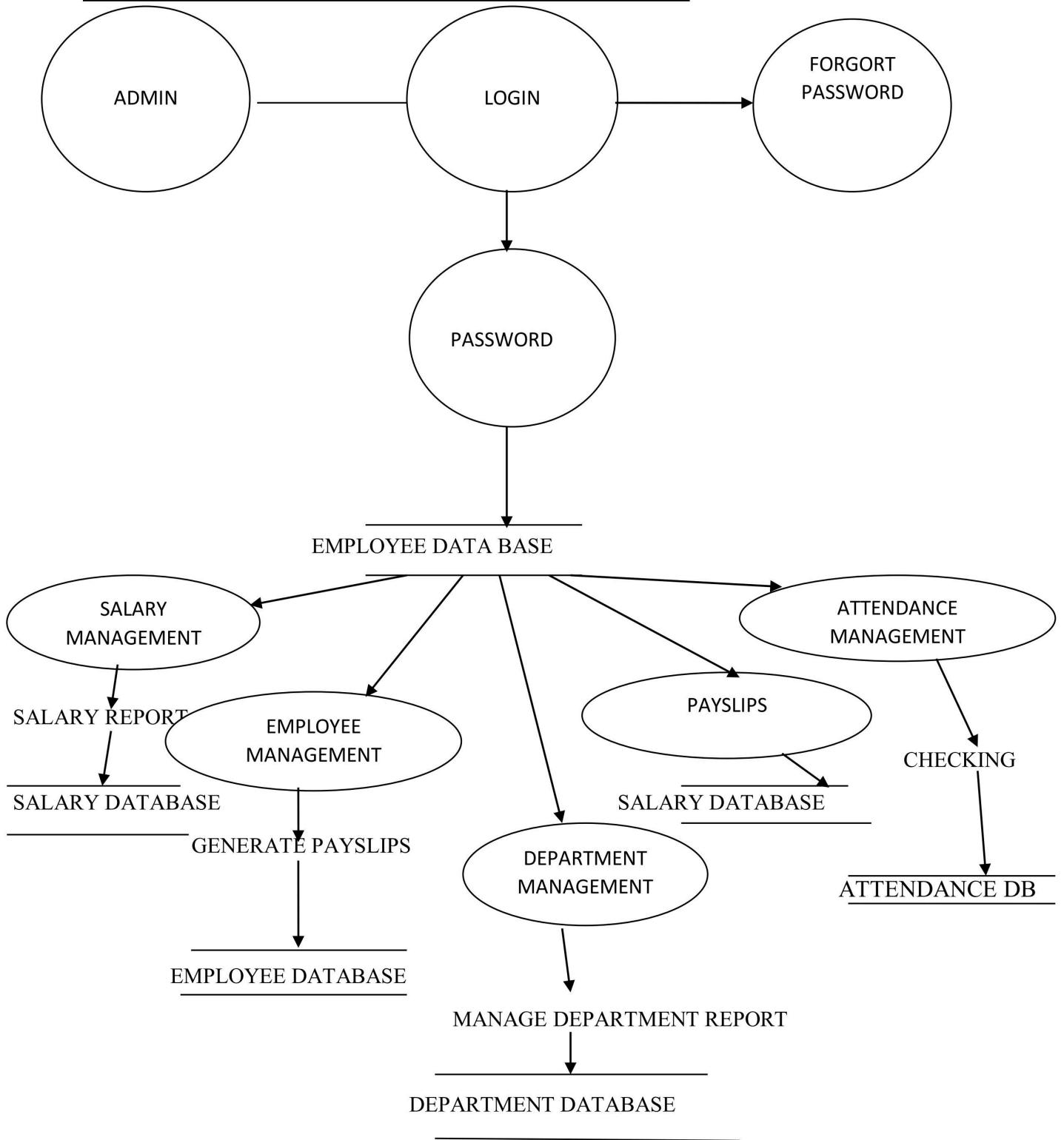
Four symbols are used in drawing dataflow diagrams, These are:



LEVEL 0 PAYROLL MANAGEMENT SYSTEMS

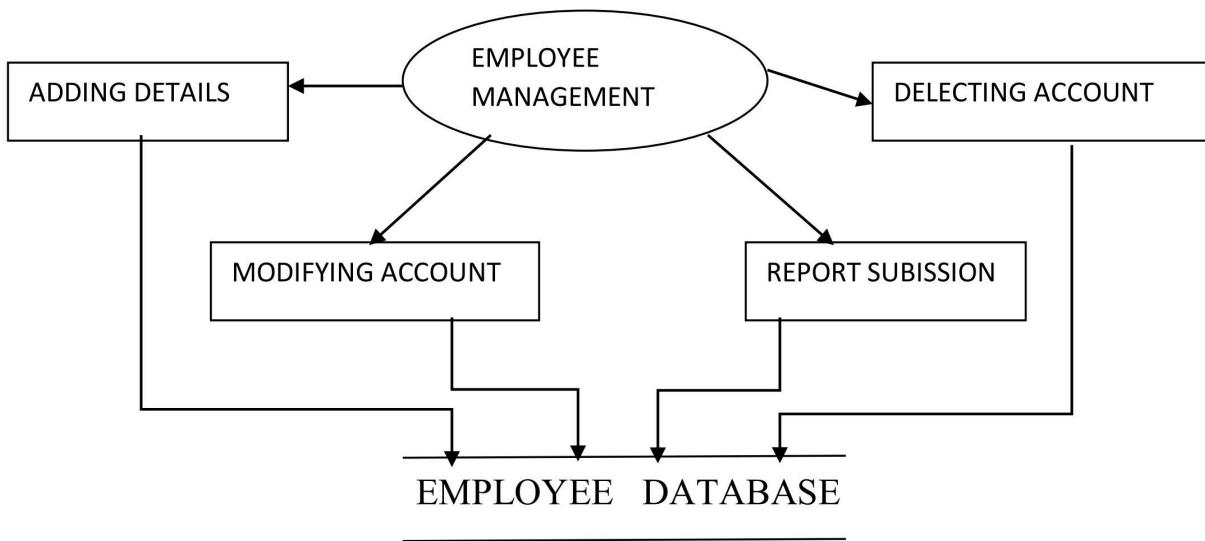


LEVEL 1 PAYROLL MANAGEMENT SYSTEM

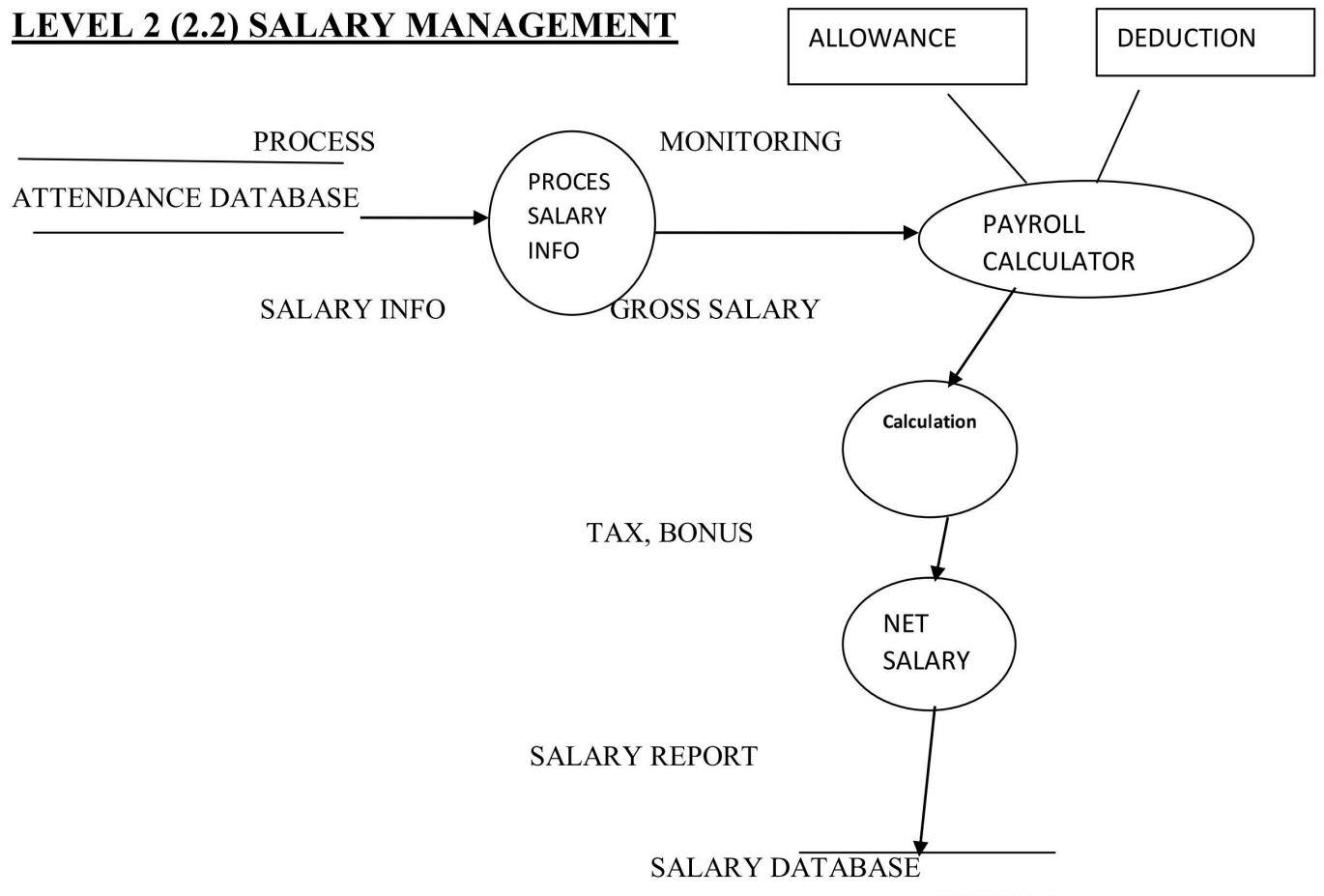


LEVEL 2(2.1) DATA FLOW DIAGRAM

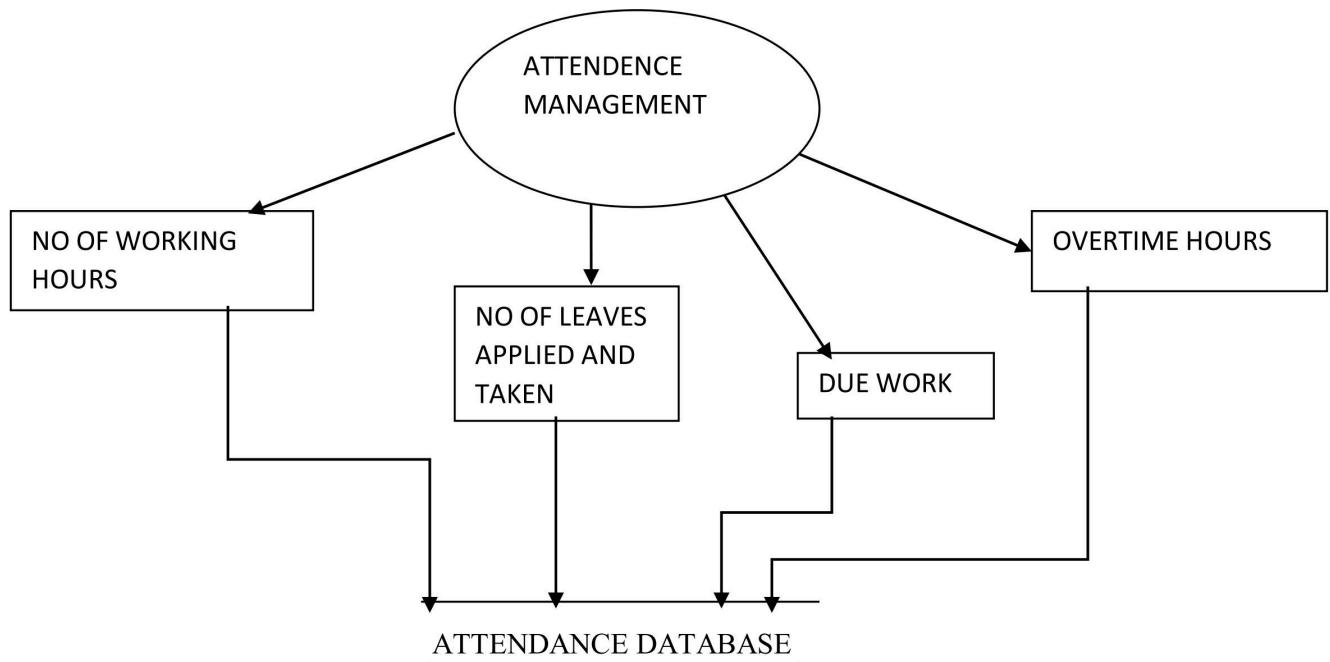
SALARY AND EMPLOYEE MANAGEMENT



LEVEL 2 (2.2) SALARY MANAGEMENT



Level 2(2.3) ATTENDANCE MANAGEMENT



PAYROLL MANAGEMENT SYSTEM

TABLES

TABLE STRUCTURE

LOGIN

<u>FIELD NAME</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
USERNAME	VARCHAR	THIS FIELD STORED THE USERNAME
PASSWORD	VARCHAR	THIS FIELD STORED THE PASSWORD

EMPLOYEE DETAILS

<u>FIELD NAME</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
USER NAME	VARCHAR	THIS FIELD CONTAINS EMPLOYEE NAME
EMPLOYEE LOGIN	NUMERIC	THIS FIELD CONTAINS EMPLOYEE LOGIN
AGE	NUMERIC	THIS FIELD STORES EMPLOYEE AGE
GENDER	VARCHAR	THIS FIELD STORES GENDER
DOB	NUMERIC	THIS FIELD STORES ACCOUNT TYPE
ADDRESS	VARCHAR	THIS FIELD STORES ADDRESS
DEPARTMENT	VARCHAR	THIS FIELD STORES WHICH DEPARTMENT IS IN EMPLOYEE
CITY	VARCHAR	THIS FIELD STORES CITY
PHONE NO	NUMERIC	THIS FIELD STORES PHONE NO
EMAIL ID	VARCHAR	THIS FIELD STORES EMAIL ID
DEPARTMENT NO	VARCHAR	FIELD STORES DEPARTMENT NO

ATTENDENCE DETAILS

<u>FIELD NAME</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
NO OF WORKING DAYS	NUMERIC	THIS FIELD DISPLAYS THE NUMBER OF WORKING DAYS
TOTAL NO. OF WORKING	NUMERIC	THIS FIELD DISPLAYS THE TOTAL NO OF WORKING DAYS OF THE EMPLOYEE
WORKED HOURS	NUMERIC	THIS DISPLAYS THE WORKING HOURS OF THE EMPLOYEE
NO OF LEAVES (SICK, CASUAL)	NUMERIC	THIS FIELD DISPLAYS THE NO OF LEAVES APPLIED
NO OF LEAVES TAKEN	NUMERIC	THIS FIELD DISPLAYS THE NO OF LEAVES TAKEN
WAGES	NUMERIC	THIS DISPLAYS THE EMPLOYEE WAGES
LEAVE DESCRIPTION	VARCHAR	THIS FIELD STORES DESCRIPTION OF LEAVE
MONTHLY LEAVE	NUMERIC	THIS FIELD STORES MONTHLY LEAVES

DEPARTMENT TABLE

<i>Field name</i>	<i>Data type</i>	<i>Description</i>
USER NAME	VARCHAR	THIS FIELD STORED THE USERNAME
LOGIN ID	VARCHAR	THIS FIELD STORED THE LOGIN ID
DEPARTMENT	VARCHAR	THIS FIELD STORED THE DEPARTMENT
DESCRIPTION	VARCHAR	FIELD STORED CONTAIN

SALARY DETAILS

<u>FIELD NAME</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
EMPLOYEE LOGIN	VARCHAR	THIS FIELD CONTAINS EMPLOYEE LOGIN
USER NAME	CHARACTER	THIS FIELD CONTAINS EMPLOYEE NAME
SALARY	NUMERIC	THIS FIELD STORES EMPLOYEE SALARY
ALLOWANCE	NUMERIC	THIS FIELD STORES EMPLOYEE ALLOWANCE
DEDUCTION	NUMERIC	THIS FIELD STORES EMPLOYEE DEDUCTION
NET SALARY	NUMERIC	THIS FIELD STORES EMPLOYEE TOTAL SALARY
BONUS	NUMERIC	THIS FIELD STORES EMPLOYEE BONUS
TAX ABLE PAY	NUMERIC	THIS FIELD STORES EMPLOYEE TAX
PENSIONABLE PAY	NUMERIC	THIS FIELD STORES EMPLOYEE PENSION
STUDENT LOAN	NUMERIC	THIS FIELD STORES EMPLOYEE LOAN
PAYDATE	NUMERIC	THIS FIELD STORES EMPLOYEE PAY DATE
OVERTIME	NUMERIC	THIS FIELD STORES EMPLOYEE OVER TIME
CITYWEIGHTING	NUMERIC	THIS FIELD STORES EMPLOYEE CITY WEIGHT
OTHER PAYMENT	NUMERIC	THIS FIELD STORES EMPLOYEE EXTRA PAY
TAX CODE	NUMERIC	GENERAL TAX CODE
TAX PERIOD	NUMERIC	GENRAL TAX DATE
NIPERIOD	NUMERIC	NETINCOME DATE

NIPAYMENT	NUMERIC	NET INCOME PAYMENT YEARLY
PAY DATE	NUMERIC	GENERAL PAY DATES TO EMPLOYEE

ER DIAGRAM

Er diagram allows us to sketch database design. ERD is a graphical tool for modelling data. It is a graphical representation of the logical structure of the database.

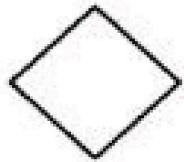
Various symbols used in ERD are:-



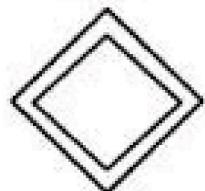
STRONG ENTITY SET



WEAK ENTITY



RELATIONSHIP



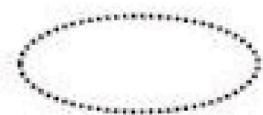
IDENTIFYING
RELATIONSHIP



SIMPLE
ATTRIBUTES



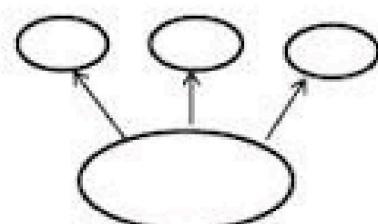
COMPOSIT
ATTRIBUTES



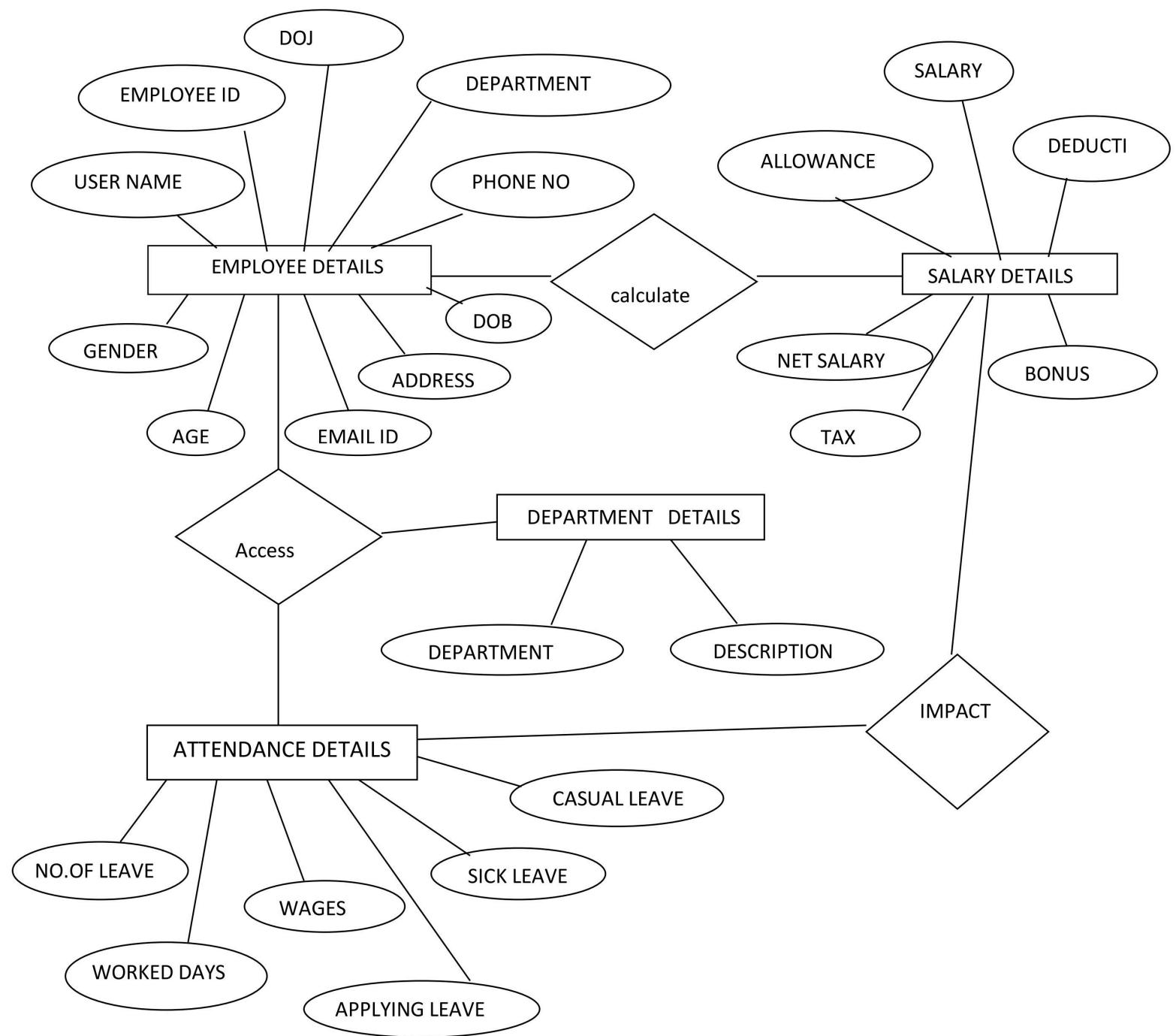
DERIVED
ATTRIBUTES



MULTIVALUED
ATTRIBUTES

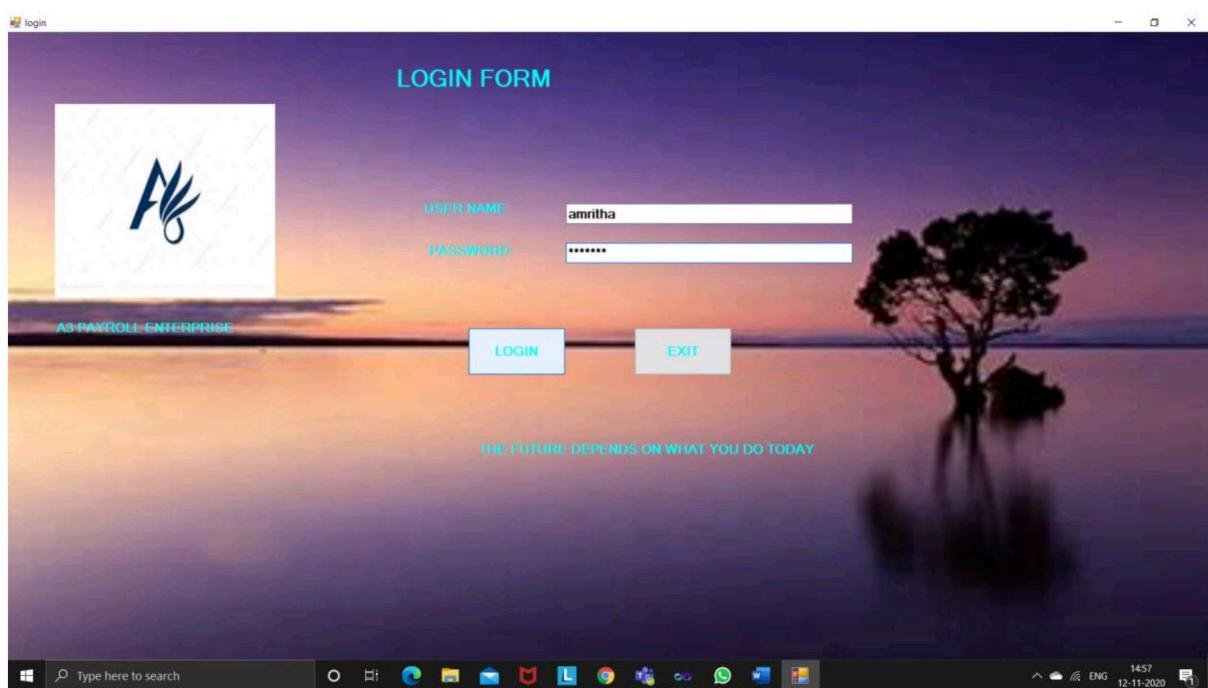
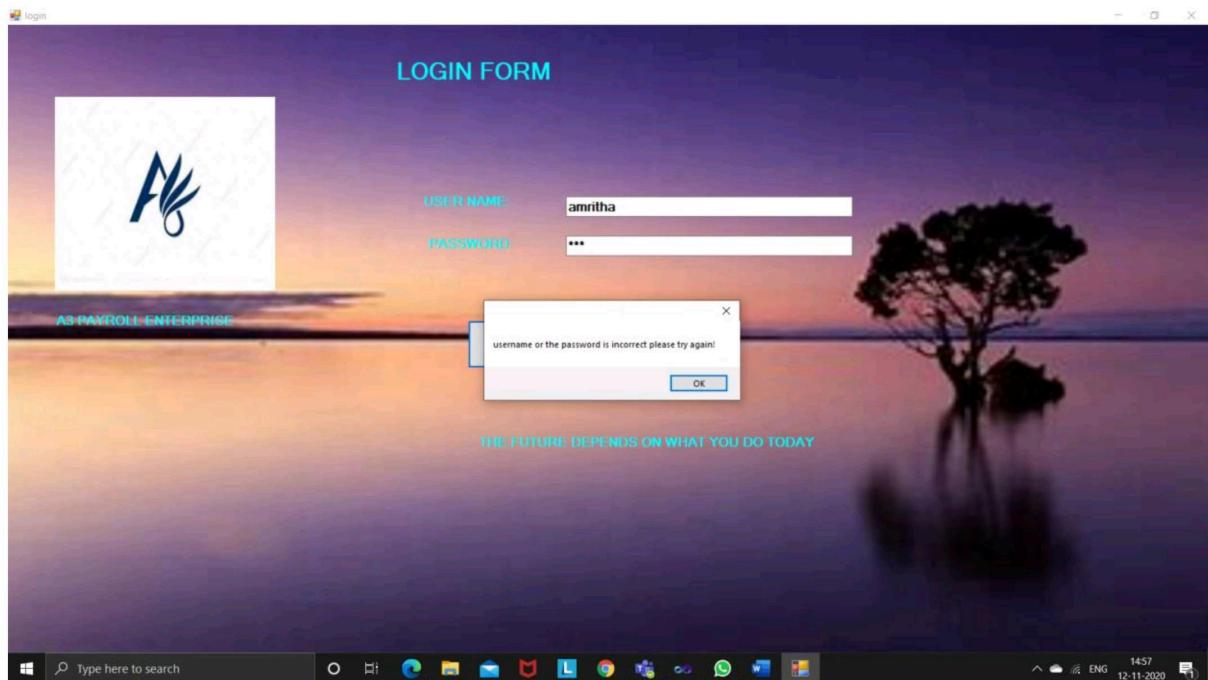


COMPOSIT
ATTRIBUTES

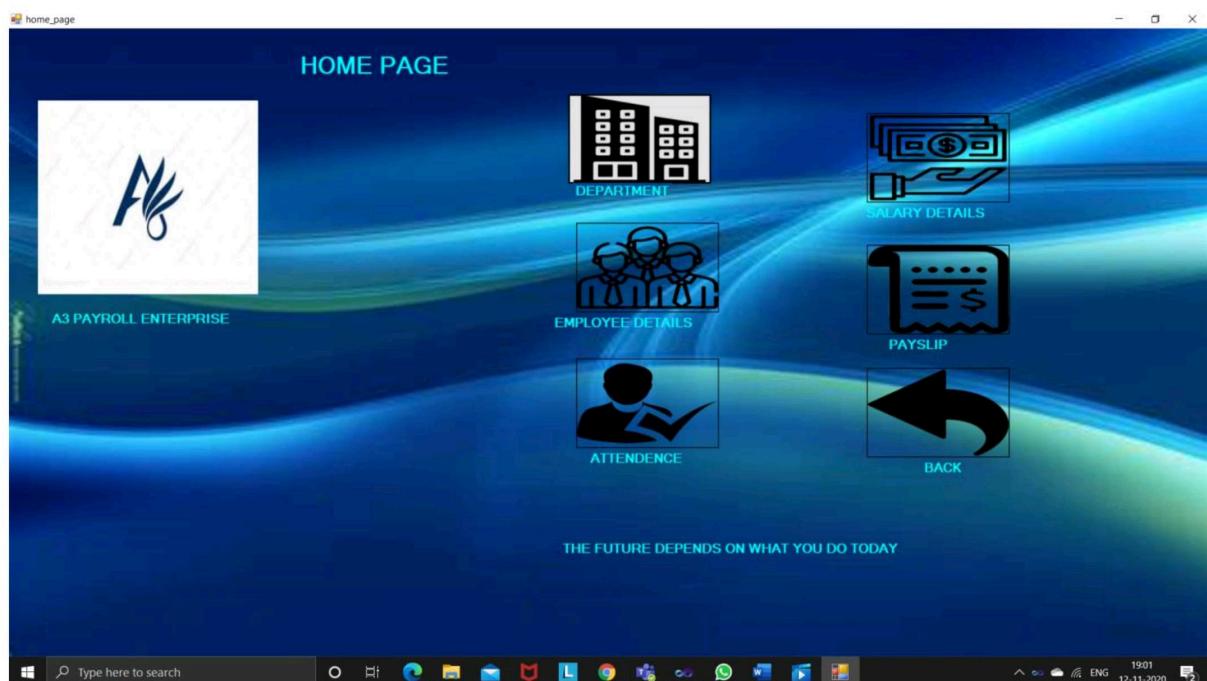
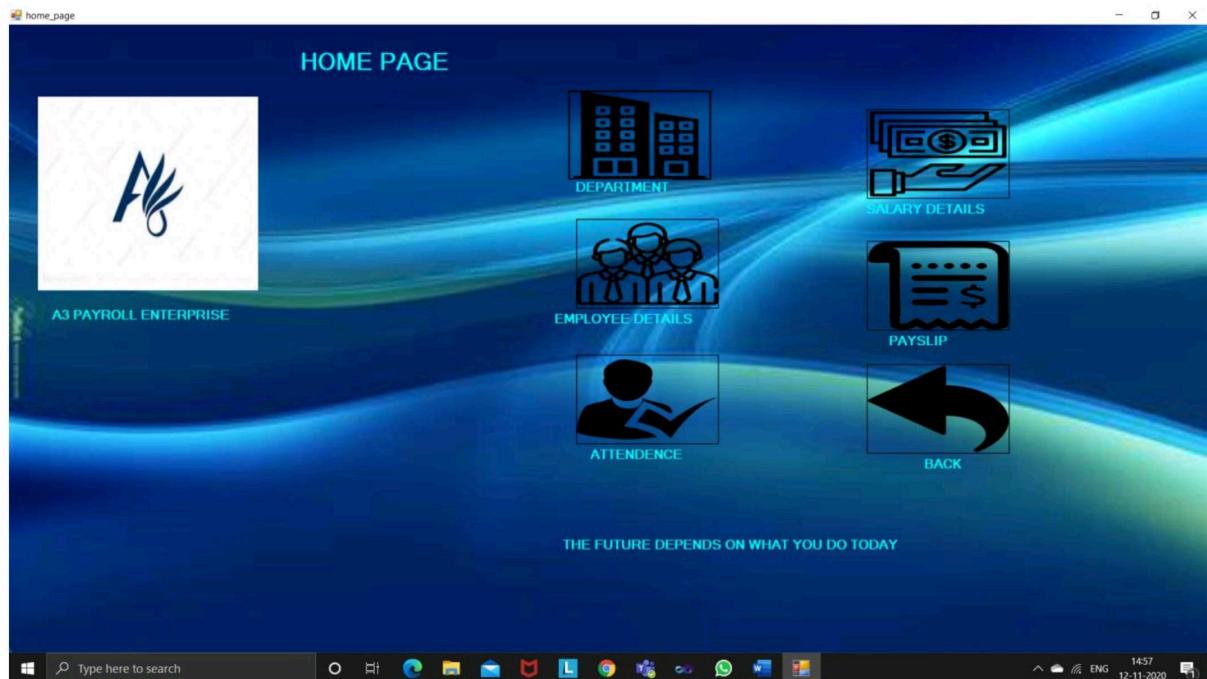


User Interface Design

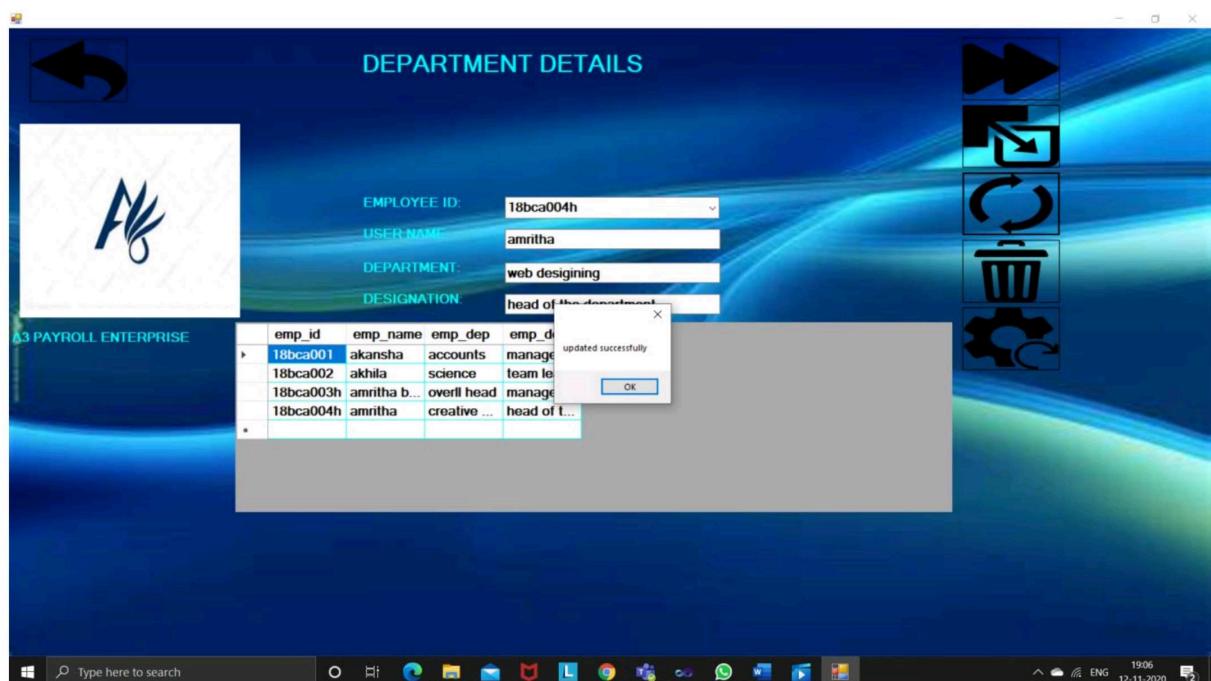
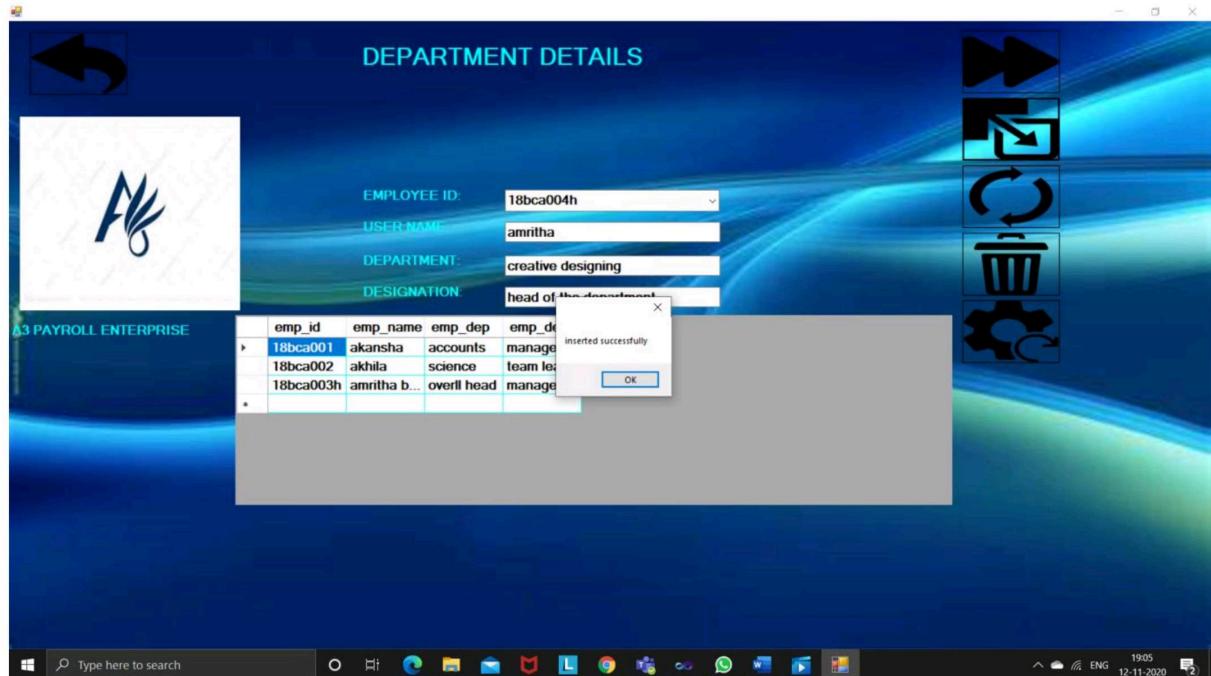
Login form

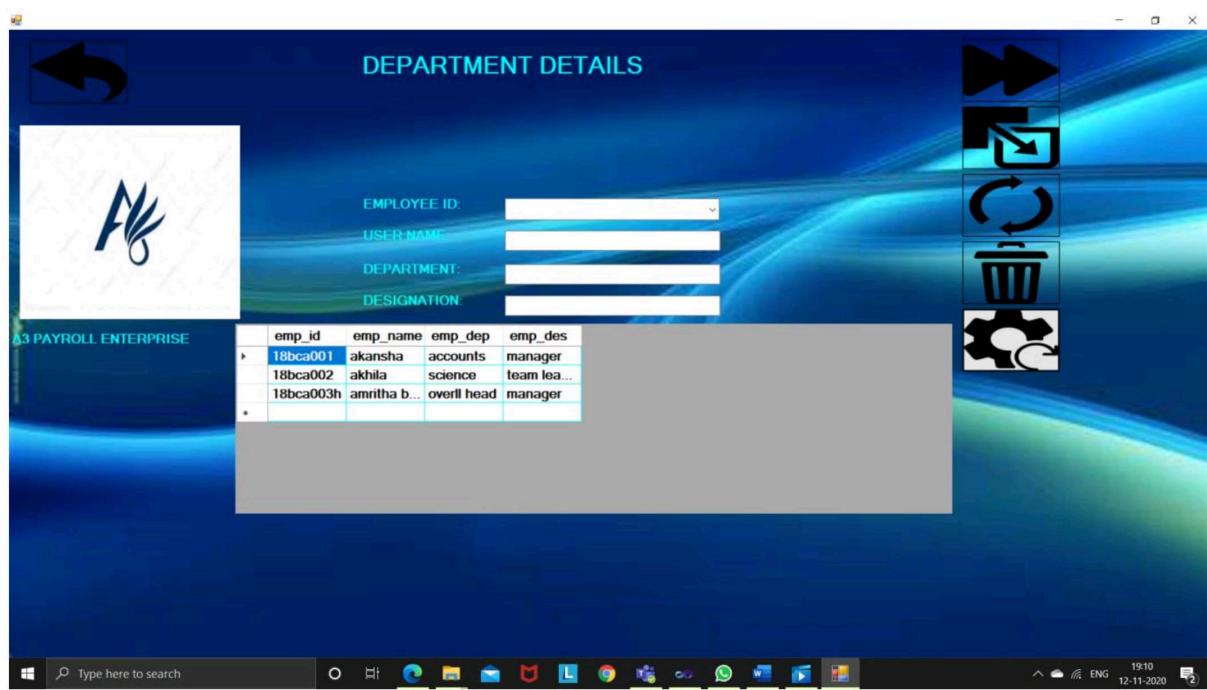
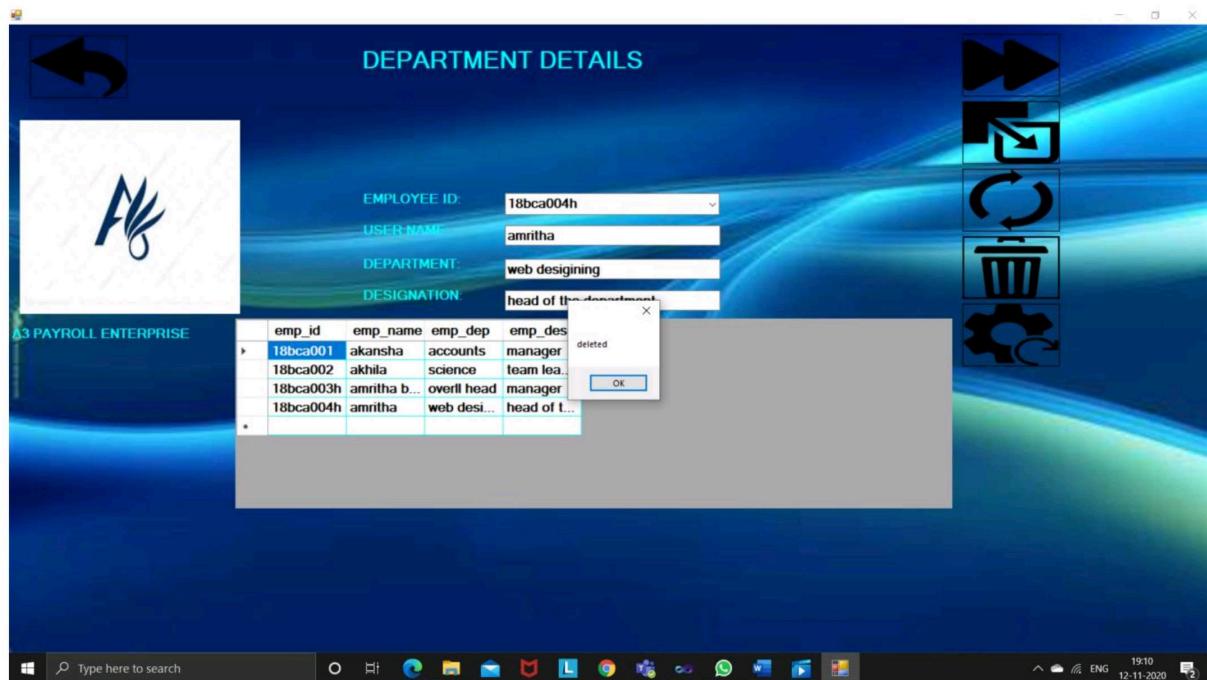


Home page

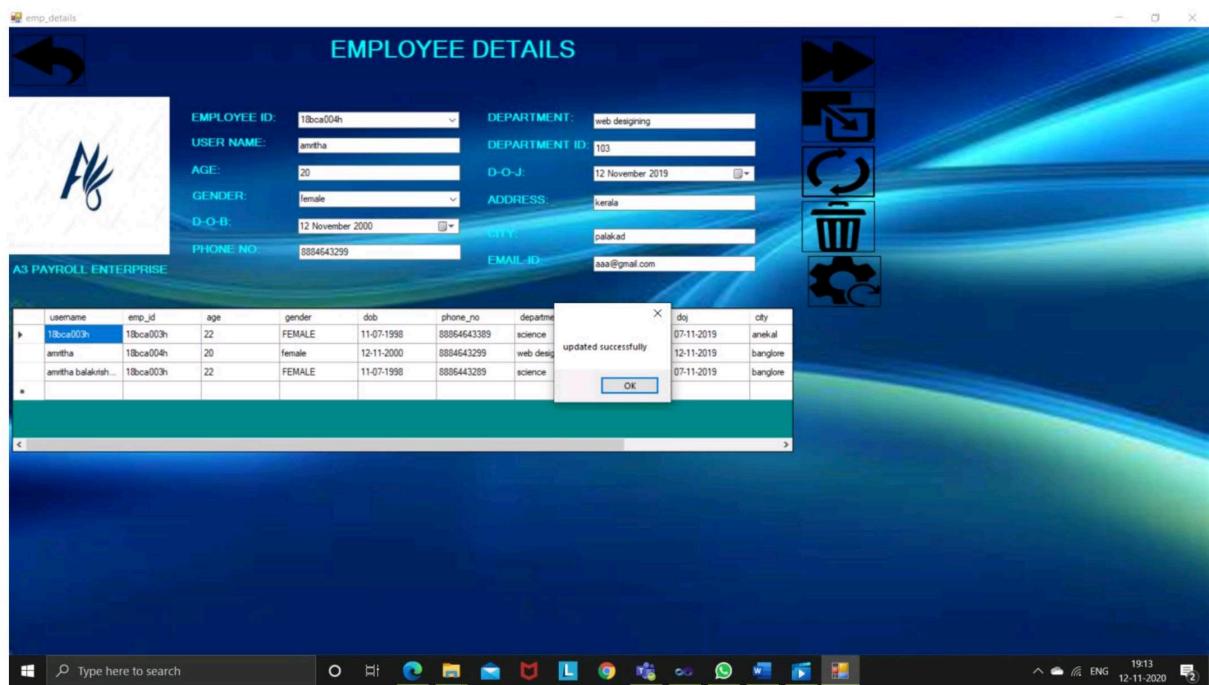
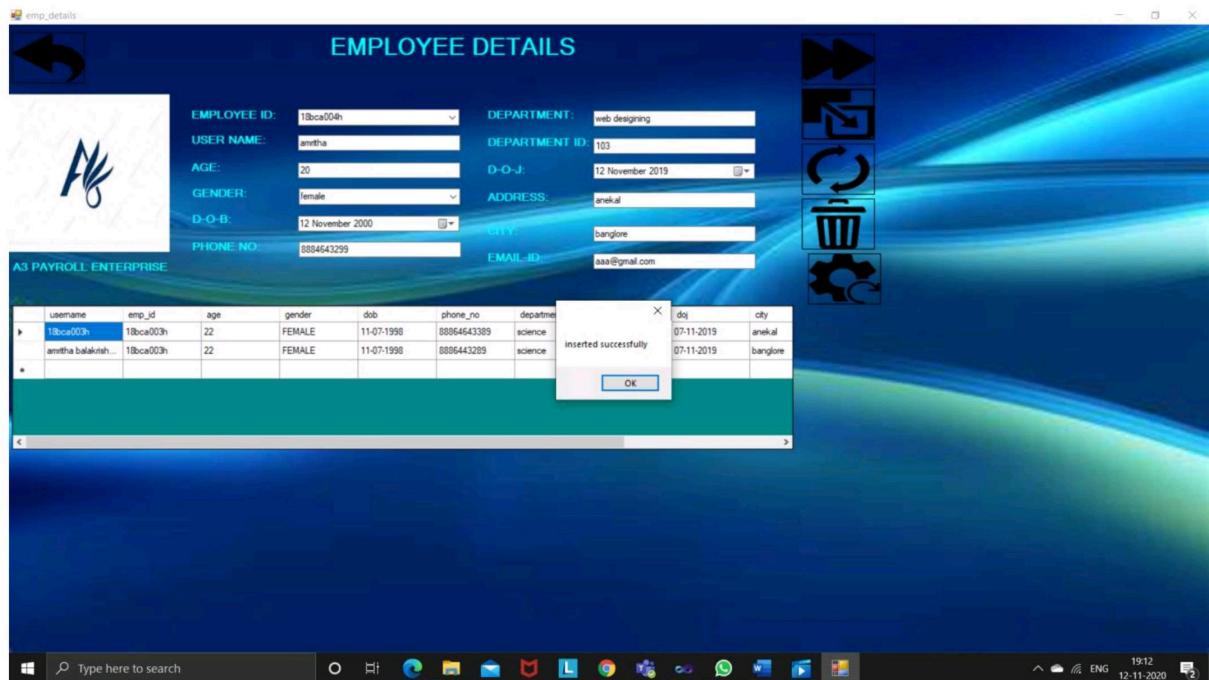


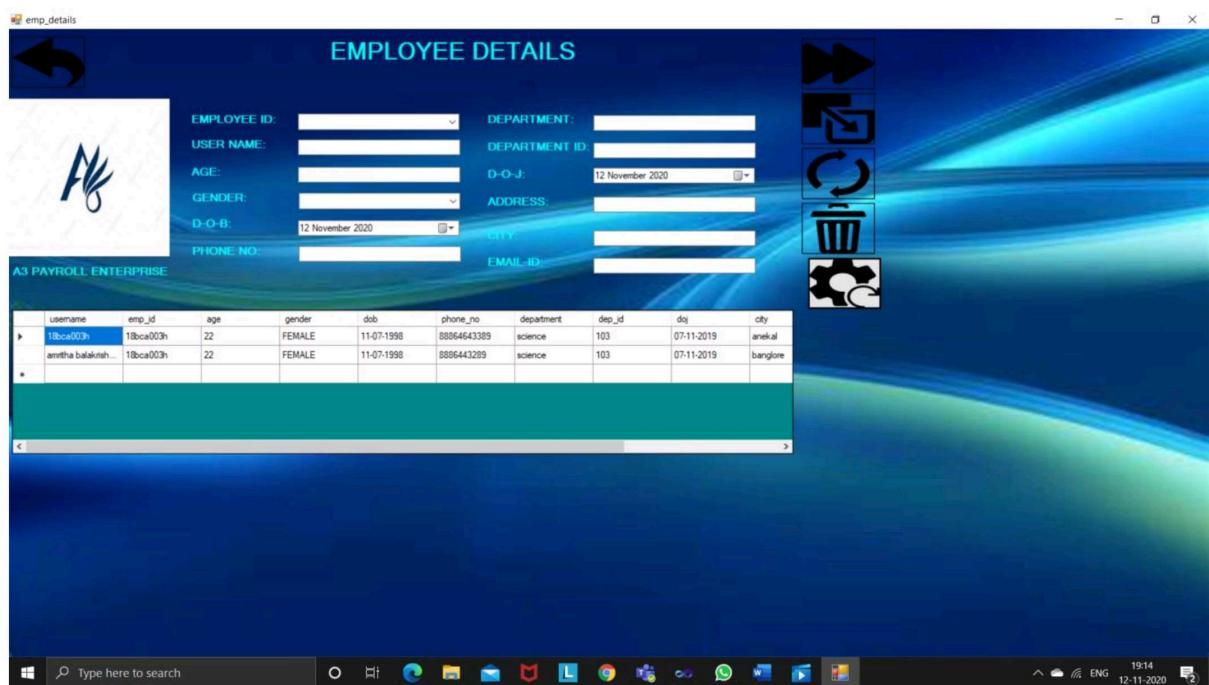
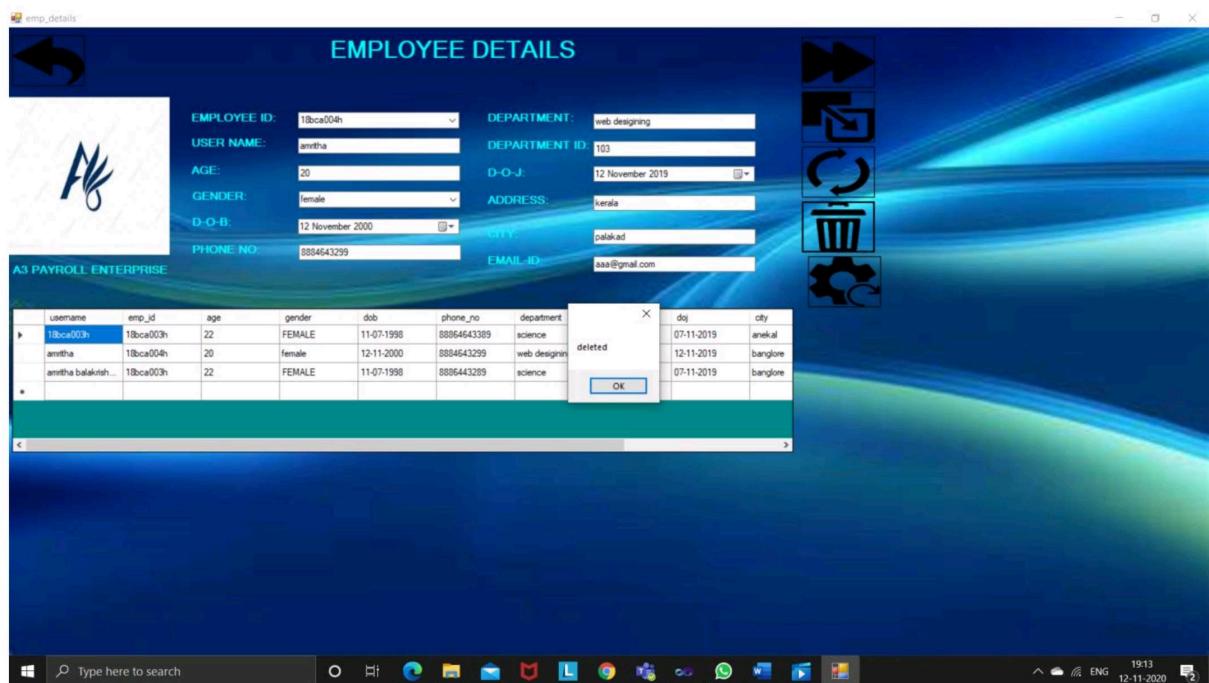
Department form



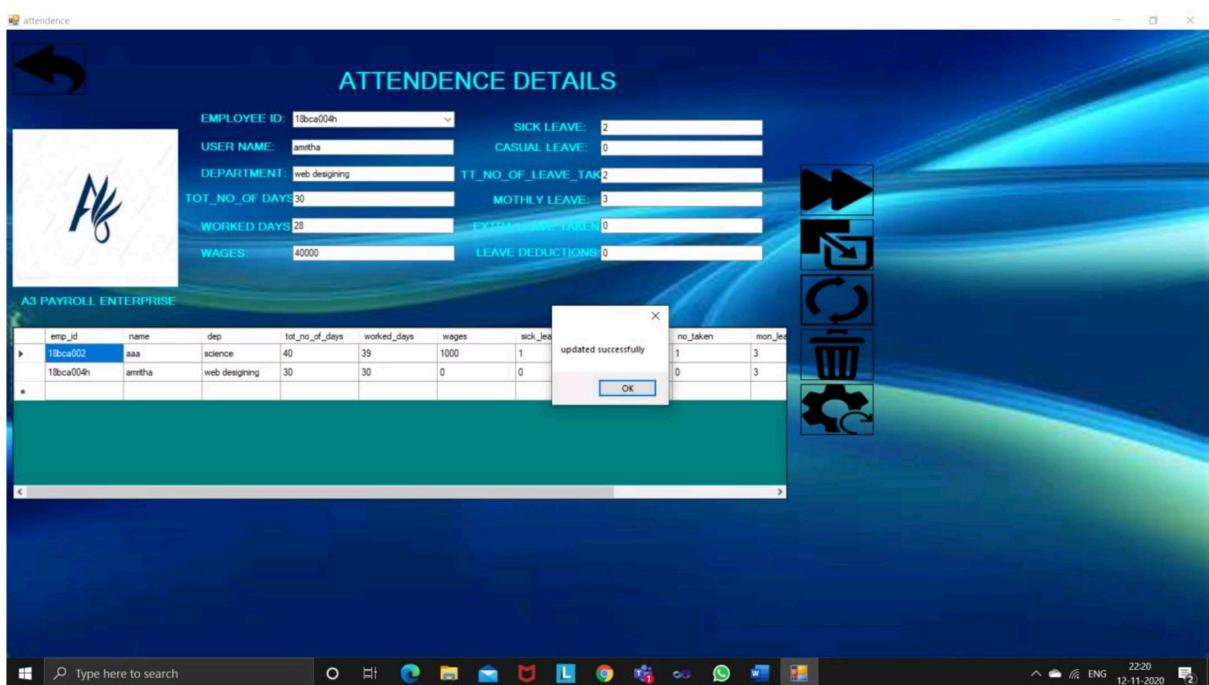
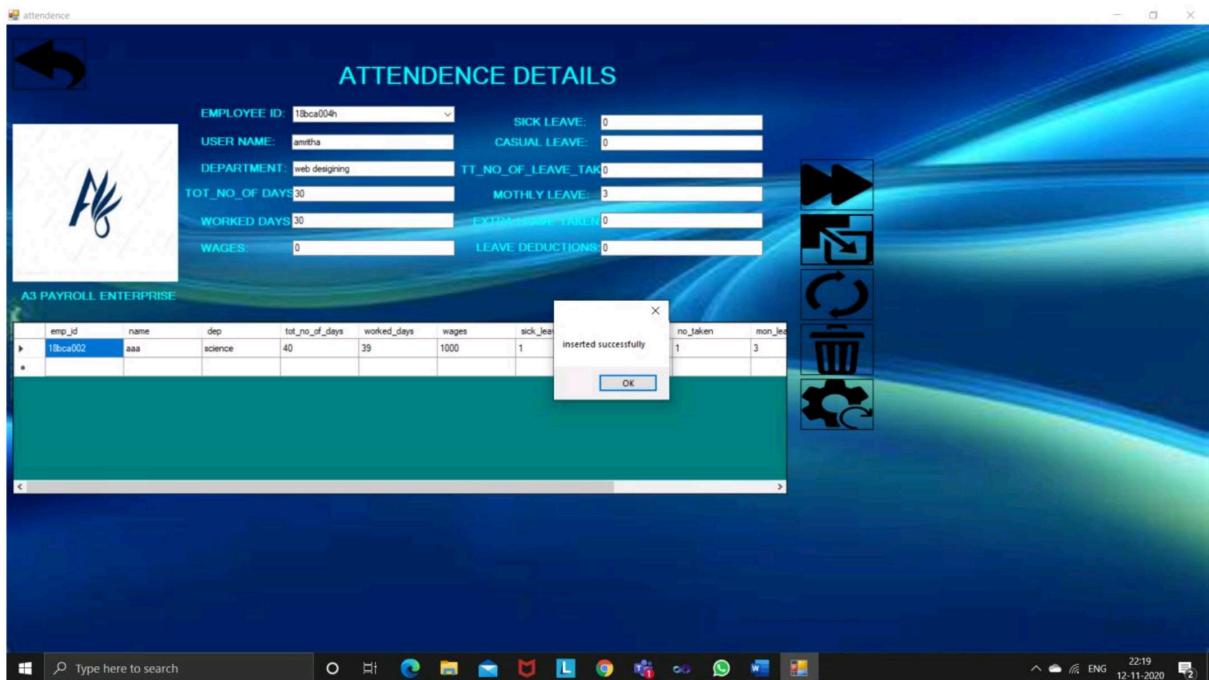


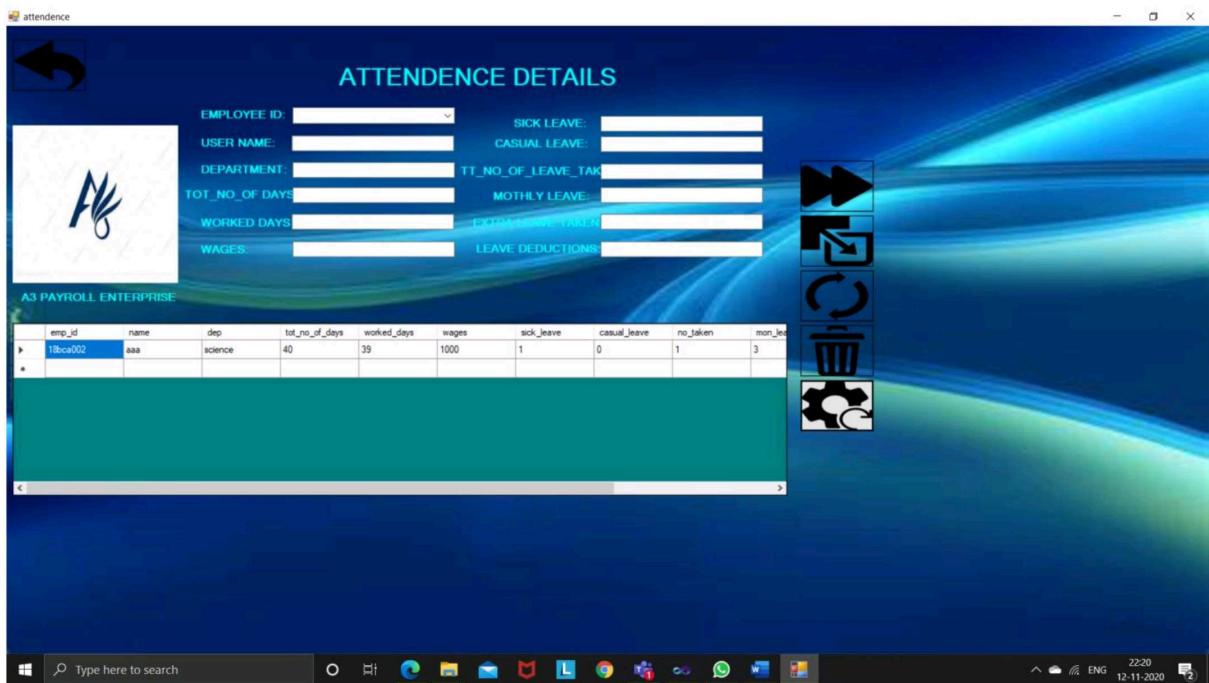
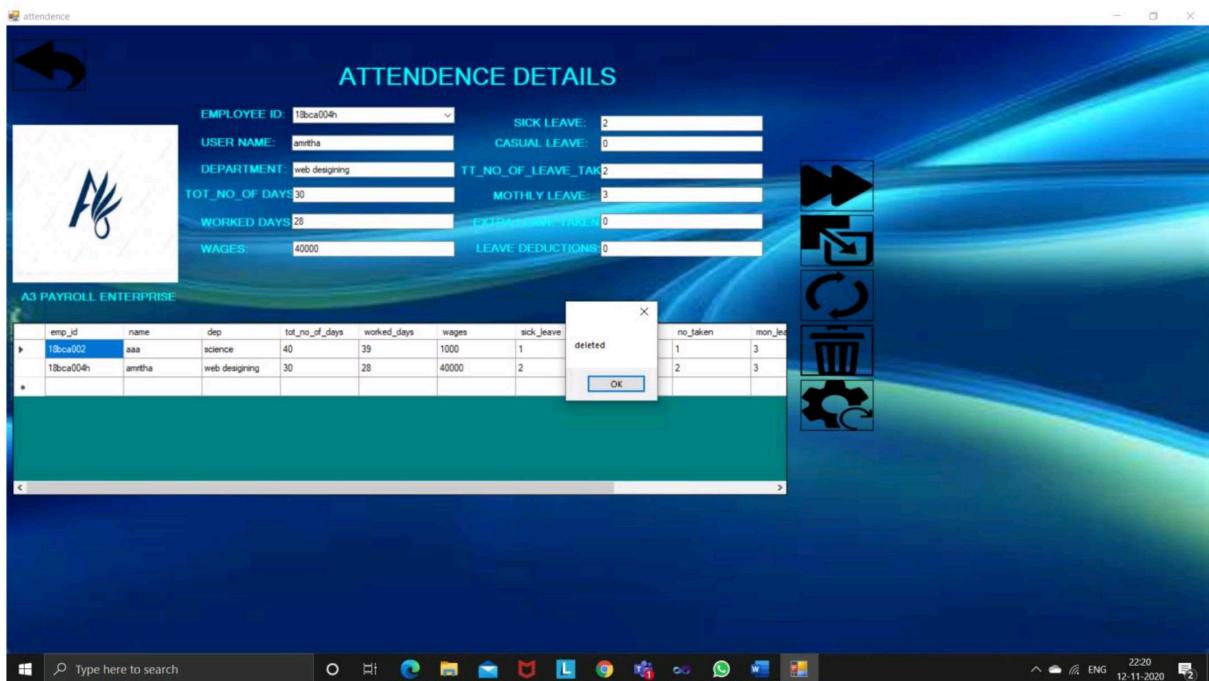
Employee details



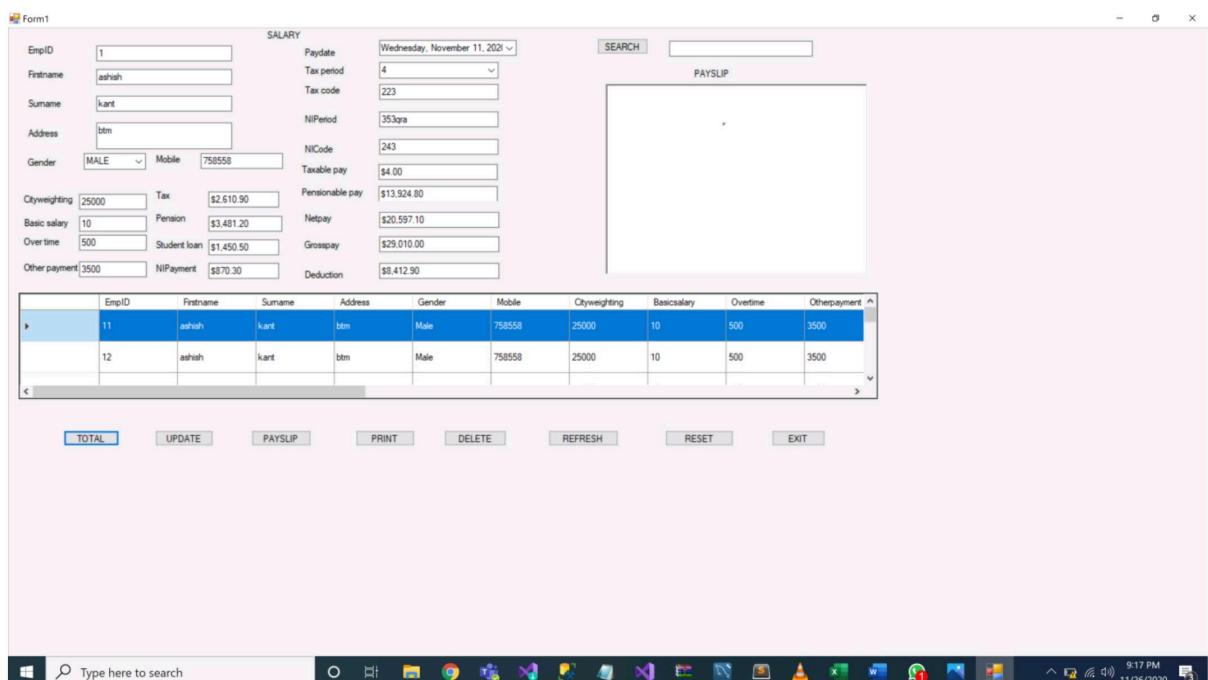
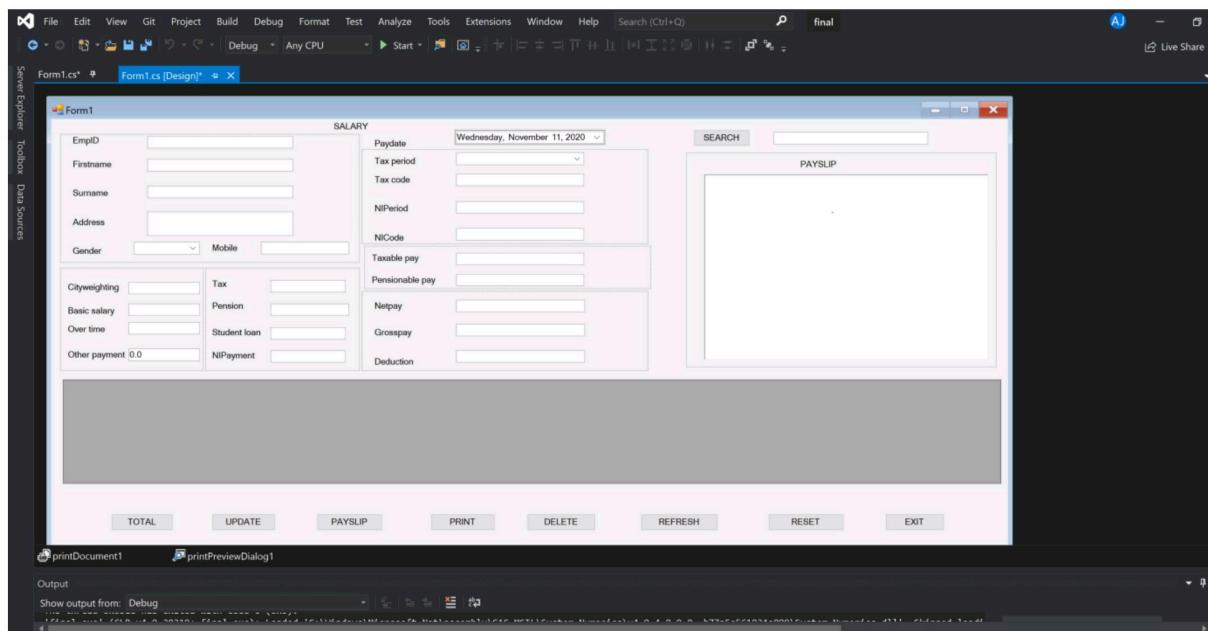


Attendance form





Salary and Payslip form



Form1

SALARY						PAYSPLIT																			
EmplID	56	Paydate	Thursday , November 26, 2021	SEARCH																					
Fname	asda	Tax period	2																						
Surname	gngd	Tax code	gngd																						
Address	gaga	NIPeriod	435342																						
Gender	MALE	NICode	fgag																						
Ctyweighting	123	Taxable pay																							
Basic salary	443	Pensionable pay																							
Overtime	355	Grosspay																							
Other payment	33	NIpayment																							
Deduction		Deduction																							
<table border="1"> <thead> <tr> <th>EmplID</th><th>Fname</th><th>Surname</th><th>Address</th><th>Gender</th><th>Mobile</th> </tr> </thead> <tbody> <tr> <td>11</td><td>ashish</td><td>kant</td><td>btm</td><td>Male</td><td>758558</td> </tr> <tr> <td>12</td><td>ashish</td><td>kant</td><td>btm</td><td>Male</td><td>758558</td> </tr> </tbody> </table>						EmplID	Fname	Surname	Address	Gender	Mobile	11	ashish	kant	btm	Male	758558	12	ashish	kant	btm	Male	758558	Employee system	
EmplID	Fname	Surname	Address	Gender	Mobile																				
11	ashish	kant	btm	Male	758558																				
12	ashish	kant	btm	Male	758558																				
						Overtime	Otherpayment ^																		
						500	3500																		
						500	3500																		

confirm if you want to exit

Yes No

TOTAL UPDATE PAYSPLIT PRINT DELETE REFRESH RESET EXIT

9:17 PM
11/26/2020

Form1

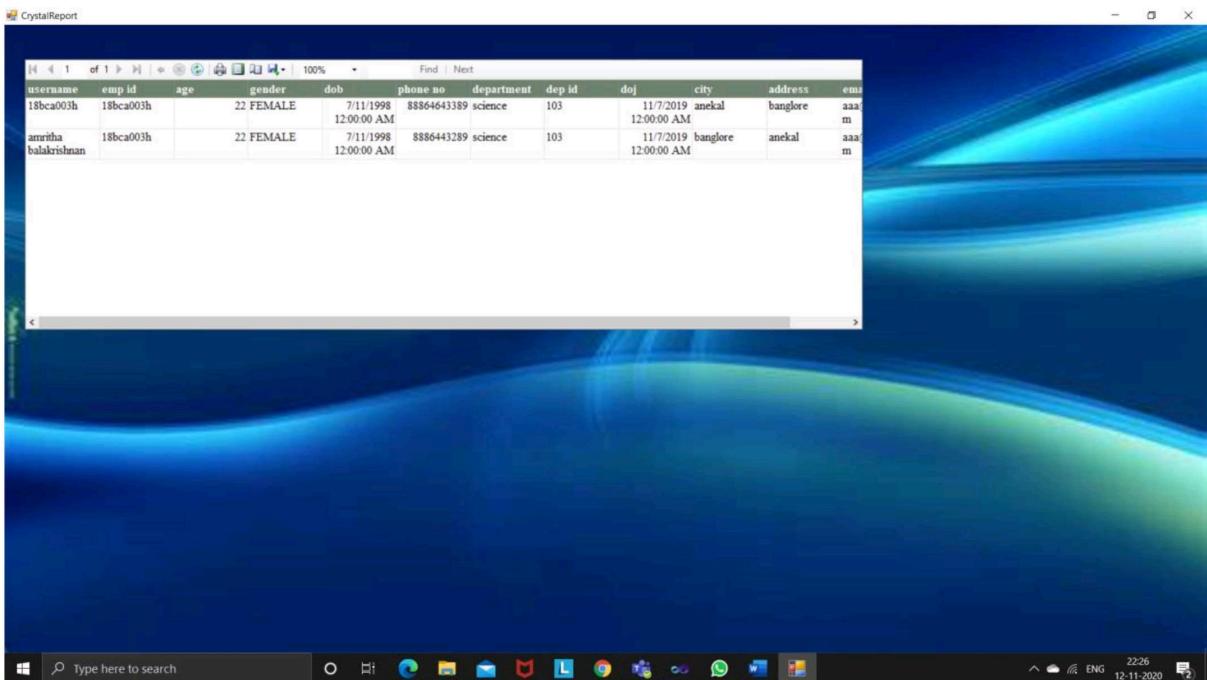
SALARY						PAYSPLIT																			
EmplID	Print preview	Page	1	SEARCH																					
Fname	asda																								
Surname	gngd																								
Address	gaga																								
Gender	MALE																								
Ctyweighting	123																								
Basic salary	443																								
Overtime	355																								
Other payment	33																								
<table border="1"> <thead> <tr> <th>EmplID</th><th>Fname</th><th>Surname</th><th>Address</th><th>Gender</th><th>Mobile</th> </tr> </thead> <tbody> <tr> <td>11</td><td>ashish</td><td>kant</td><td>btm</td><td>Male</td><td>758558</td> </tr> <tr> <td>12</td><td>ashish</td><td>kant</td><td>btm</td><td>Male</td><td>758558</td> </tr> </tbody> </table>						EmplID	Fname	Surname	Address	Gender	Mobile	11	ashish	kant	btm	Male	758558	12	ashish	kant	btm	Male	758558	Employee system	
EmplID	Fname	Surname	Address	Gender	Mobile																				
11	ashish	kant	btm	Male	758558																				
12	ashish	kant	btm	Male	758558																				
						Overtime	Otherpayment ^																		
						500	3500																		
						500	3500																		

Print preview

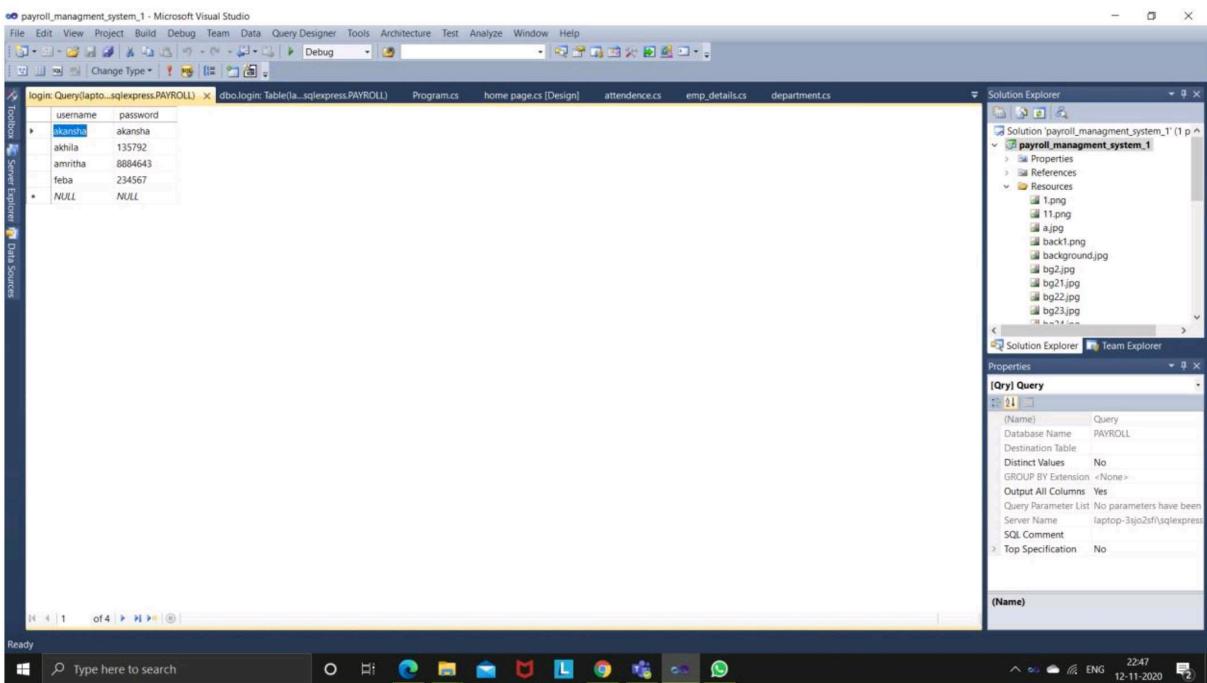
TOTAL UPDATE PAYSPLIT PRINT DELETE REFRESH RESET EXIT

9:18 PM
11/26/2020

CRYSTAL REPORT



Login database



Department database

The screenshot shows the Microsoft Visual Studio interface for a project named 'payroll_management_system_1'. The 'Solution Explorer' pane on the right lists files like 'Program.cs', 'home_page.cs [Design]', 'attendance.cs', and 'emp_details.cs'. The 'Properties' pane shows settings for a query named 'Query1' with a database name of 'PAYROLL'. The 'Toolbox' and 'Server Explorer' panes are also visible. In the center, there are two query results windows. The top window, titled 'depart: Query(laptop_sqlexpress.PAYROLL)', displays data from the 'EMPLOYEE' table:

emp_id	emp_name	emp_dep	emp_des
18bc001	akansha	accounts	manager
18bc002	akhila	science	team leader
18bc003h	amritha bal...	overall head	manager
NULL	NULL	NULL	NULL

The bottom window, titled 'login: Query(laptop_sqlexpress.PAYROLL)', displays data from the 'EMPLOYEE' table:

username	emp_id	age	gender	dob	phone_no	department	dep_id	doj	city	address	email_id
18bc003h	18bc003h	22	FEMALE	11-07-1998	88864643389	science	103	07-11-2019	anekal	banglore	aaa@gmail...
amritha bal...	18bc003h	22	FEMALE	11-07-1998	8886443289	science	103	07-11-2019	banglore	anekal	aaa@gmail...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Employee details

The screenshot shows the Microsoft Visual Studio interface for the same project. The 'Solution Explorer' pane on the right lists files like 'Program.cs', 'attendance.cs', and 'emp_details.cs'. The 'Properties' pane shows settings for a query named 'Query1' with a database name of 'PAYROLL'. The 'Toolbox' and 'Server Explorer' panes are also visible. In the center, there are two query results windows. The top window, titled 'employee: Query(laptop_sqlexpress.PAYROLL)', displays data from the 'EMPLOYEE' table:

username	emp_id	age	gender	dob	phone_no	department	dep_id	doj	city	address	email_id
18bc003h	18bc003h	22	FEMALE	11-07-1998	88864643389	science	103	07-11-2019	anekal	banglore	aaa@gmail...
amritha bal...	18bc003h	22	FEMALE	11-07-1998	8886443289	science	103	07-11-2019	banglore	anekal	aaa@gmail...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The bottom window, titled 'login: Query(laptop_sqlexpress.PAYROLL)', displays data from the 'EMPLOYEE' table:

username	emp_id	age	gender	dob	phone_no	department	dep_id	doj	city	address	email_id
18bc003h	18bc003h	22	FEMALE	11-07-1998	88864643389	science	103	07-11-2019	anekal	banglore	aaa@gmail...
amritha bal...	18bc003h	22	FEMALE	11-07-1998	8886443289	science	103	07-11-2019	banglore	anekal	aaa@gmail...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Attendance details and salary Payslip database

The screenshot shows the Microsoft Visual Studio interface. In the center, there is a 'Query Results' window titled 'leave: Query(laptop...sqlexpress.PAYOUT)'. It displays a table with columns: emp_id, name, dep, tot_no_of_, worked_days, wages, sick_leave, casual_leave, no_taken, mon_leave, extra_leave, and leave_ded. The data shows two rows: one for 'aaa' (dep: science) and one for 'amitha' (dep: science). The 'Solution Explorer' window on the right shows a project named 'payroll_management_system_1' with various files like 'background.jpg' and 'b1.png' under the 'Resources' folder.

The screenshot shows the MySQL Workbench interface. In the center, there is a 'Result Grid' window displaying the results of a query: 'SELECT * FROM employee.employeedb'. The results show multiple rows of data for employees, including columns like Otherpayment, Tax, Pension, Studentloan, NPAYment, Taxperiod, Taxcode, NPeriod, Taxablepay, Netpay, Grosspay, Deduction, Paydate, and Netpay. The 'Navigator' pane on the left shows the schema 'employeedb' with tables like 'EmployeeCom' and 'Cityweighting'. The bottom status bar indicates the date as 12-11-2020 and the time as 22:50.

Code Design

Login form

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace payroll_managment_system_1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection("Data Source=LAPTOP-
3SJO2SFI\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True");
        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void textBox2_TextChanged(object sender, EventArgs e)
        {
        }
        private void Form1_Load(object sender, EventArgs e)
        {
        }
        private void button1_Click(object sender, EventArgs e)
        {
            String query = "select * from login where username = '" + txtUserName.Text.Trim() +
            "' and password = '" + txtEmpPassword.Text.Trim() + "' ";
            SqlDataAdapter sda = new SqlDataAdapter(query, con);
            DataTable dt = new DataTable();
            sda.Fill(dt);
            if (dt.Rows.Count == 1)
            {
                this.Hide();
                home_page ss = new home_page();
                ss.Show();
            }
            else
            {
                MessageBox.Show("username or the password is incorrect please try again!");
            }
        }
    }
}
```

Home page form

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

namespace payroll_managment_system_1
{
    public partial class home_page : Form
    {
        public home_page()
        {
            InitializeComponent();
        }

        private void button6_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Hide();
            department dep = new department();
            dep.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Hide();
            emp_details emp = new emp_details();
            emp.Show();
        }

        private void button5_Click(object sender, EventArgs e)
        {
            this.Hide();
            attendence att = new attendence();
            att.Show();
        }
    }
}
```

Department form

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace payroll_managment_system_1
{
    public partial class department : Form
    {
        public department()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection("Data Source=LAPTOP-
3SJO2SFI\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True");
        private void department_Load(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("select (emp_id) from depart", con);
            SqlDataReader dr = cmd.ExecuteReader();
            while (dr.Read())
            {
                comboBox1.Items.Add(dr.GetValue(0).ToString());
            }
            dr.Close();
            con.Close();
            Refresh();
        }
        void Refresh()
        {
            con.Open();
            SqlDataAdapter ada = new SqlDataAdapter("select * from depart", con);
            DataSet ds = new DataSet();
            ada.Fill(ds, "depart");
            dataGridView1.DataSource = ds;
            dataGridView1DataMember = "depart";
            con.Close();
        }
        private void button4_Click(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("insert into depart
(emp_id,emp_name,emp_dep,emp_des) values ('" + comboBox1.Text + "','" +
txtEmpName.Text + "','" + txtEmpDep.Text + "','" + txtEmpDesign.Text + "')", con);
```

```

        cmd.ExecuteNonQuery();
        MessageBox.Show("inserted successfully");
        con.Close();
        Refresh();
    }

private void button3_Click(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("update depart set emp_name = '" +
txtEmpName.Text + "'", emp_dep = "'" + txtEmpDep.Text + "'", emp_des = "'" +
txtEmpDesign.Text + "' where emp_id=''" + comboBox1.Text + "' ", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("updated successfully");
    con.Close();
    Refresh();
}
private void button2_Click(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("delete from depart where emp_id=''" +
comboBox1.Text + "' ", con);
    MessageBox.Show("deleted");
    cmd.ExecuteNonQuery();
    con.Close();
    Refresh();
}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("select * from depart where emp_id ='" +
comboBox1.SelectedItem.ToString() + "' ", con);
    SqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        comboBox1.Text = dr.GetValue(0).ToString();
        txtEmpName.Text = dr.GetValue(1).ToString();
        txtEmpDep.Text = dr.GetValue(2).ToString();
        txtEmpDesign.Text = dr.GetValue(3).ToString();
    }
    dr.Close();
    con.Close();
}
private void button6_Click(object sender, EventArgs e)
{
    this.Close();
}
private void button1_Click(object sender, EventArgs e)
{
    this.Hide();
    emp_details emp = new emp_details();

```

```
        emp.Show();
    }
private void button5_Click(object sender, EventArgs e)
{
    comboBox1.Text = string.Empty;
    txtEmpName.Clear();
    txtEmpDep.Clear();
    txtEmpDesign.Clear();
}
}
```

Employee form

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace payroll_managment_system_1
{
    public partial class emp_details : Form
    {
        public emp_details()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection("Data Source=LAPTOP-
3SJO2SFI\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True");
        private void button6_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = new SqlCommand(" insert into employee
(emp_id,username,age,gender,dob,phone_no,department,dep_id,doj,city,address,email_id)
values('" + comboBox2.Text + "' ,'" + txtEmpName.Text + "' ,'" + txtEmpAge.Text + "' ,'" +
comboBox1.Text + "' ,'" + dateTimePicker1.Text + "' ,'" + txtEmpContact.Text + "' ,'" +
txtEmpDep.Text + "' ,'" + txtDepid.Text + "' ,'" + dateTimePicker2.Text + "' ,'" +
txtEmpCity.Text + "' ,'" + txtEmpAdress.Text + "' ,'" + txtEmpEmail.Text + "' ) ", con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("inserted successfully");
            con.Close();
            Refresh();
        }
        private void button3_Click(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("update employee set username = '" +
txtEmpName.Text + "' , age = '" + txtEmpAge.Text + "' , gender = '" + comboBox1.Text + "' ,
dob = '" + dateTimePicker1.Text + "' , phone_no = '" + txtEmpContact.Text + "' , department =
'" + txtEmpDep.Text + "' , dep_id = '" + txtDepid.Text + "' , doj = '" + dateTimePicker2.Text +
"' , address = '" + txtEmpAdress.Text + "' where emp_id = '" + comboBox2.Text + "' ",
con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("updated successfully");
        }
    }
}
```

```

        con.Close();
        Refresh();
    }

private void button2_Click(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("delete from employee where emp_id='" +
comboBox2.Text + "' ", con);
    MessageBox.Show("deleted");
    cmd.ExecuteNonQuery();
    con.Close();
    Refresh();
}
private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("select * from employee where emp_id='" +
comboBox2.SelectedItem.ToString() + "' ", con);
    SqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        comboBox2.Text = dr.GetValue(0).ToString();
        txtEmpName.Text = dr.GetValue(1).ToString();
        txtEmpAge.Text = dr.GetValue(2).ToString();
        comboBox1.Text = dr.GetValue(3).ToString();
        dateTimePicker1.Text = dr.GetValue(4).ToString();
        txtEmpContact.Text = dr.GetValue(5).ToString();
        txtEmpDep.Text = dr.GetValue(6).ToString();
        txtDepid.Text = dr.GetValue(7).ToString();
        dateTimePicker2.Text = dr.GetValue(8).ToString();
        txtEmpAdress.Text = dr.GetValue(9).ToString();
        txtEmpCity.Text = dr.GetValue(10).ToString();
        txtEmpEmail.Text = dr.GetValue(11).ToString();
    }
    dr.Close();
    con.Close();
}
private void emp_details_Load(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("select (emp_id) from employee", con);
    SqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        comboBox2.Items.Add(dr.GetValue(0).ToString());
    }
    dr.Close();
    con.Close();
    Refresh();
}

```

```

    }

void Refresh()
{
    con.Open();
    SqlDataAdapter ada = new SqlDataAdapter("select * from employee", con);
    DataSet ds = new DataSet();
    ada.Fill(ds, "employee");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "employee";
    con.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    this.Hide();
    attendence att = new attendence();
    att.Show();
    Refresh();
}

private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
}

private void button5_Click(object sender, EventArgs e)
{
    comboBox2.Text = string.Empty;
    txtEmpName.Clear();
    txtEmpAge.Clear();
    comboBox1.Text = string.Empty;
    dateTimePicker1.Text = string.Empty;
    txtEmpContact.Clear();
    txtEmpDep.Clear();
    txtDepid.Clear();
    dateTimePicker2.Text = string.Empty;
    txtEmpAdress.Clear();
    txtEmpCity.Clear();
    txtEmpEmail.Clear();
}
}
}

```

Attendance form

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

namespace payroll_managment_system_1
{
    public partial class attendance : Form
    {
        public attendance()
        {
            InitializeComponent();
        }
        SqlConnection con = new SqlConnection("Data Source=LAPTOP-
3SJO2SFI\SQLEXPRESS;Initial Catalog=PAYROLL;Integrated Security=True");
        private void button6_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("insert into leave
(emp_id,name,dep,tot_no_of_days,worked_days,wages,sick_leave,casual_leave,no_taken,mo
n_leave,extra_leave,leave_ded) values ( '" + comboBox1.Text + "' , '" + txtEmpName.Text +
"' , '" + txtEmpDep.Text + "' , '" + txtEmpTot.Text + "' , '" + txtEmpWork.Text + "' ,
'" + txtEmpWage.Text + "' , '" + txtEmpSick.Text + "' , '" + txtEmpCasual.Text + "' ,
'" + txtEmpNo.Text + "' , '" + txtEmpMon.Text + "' , '" + txtEmpExtra.Text + "' ,
'" + txtEmpDed.Text + "' ) ", con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("inserted successfully");
            con.Close();
            Refresh();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("update leave set name = '" +
txtEmpName.Text + "' , dep = '" + txtEmpDep.Text + "' , tot_no_of_days = '" +
txtEmpTot.Text + "' , worked_days = '" + txtEmpWork.Text + "' , wages = '" +
txtEmpWage.Text + "' , sick_leave = '" + txtEmpSick.Text + "' , casual_leave = '" +
txtEmpCasual.Text + "' , no_taken = '" + txtEmpNo.Text + "' , mon_leave = '" +
```

```

txtEmpMon.Text + " , extra_leave = " + txtEmpExtra.Text + " , leave_ded = " +
txtEmpDed.Text + " where emp_id="" + comboBox1.Text + " ", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("updated successfully");
    con.Close();
    Refresh();
}
private void attendance_Load(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("select (emp_id) from leave", con);
    SqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        comboBox1.Items.Add(dr.GetValue(0).ToString());
    }
    dr.Close();
    con.Close();
    Refresh();
}
void Refresh()
{
    con.Open();
    SqlDataAdapter ada = new SqlDataAdapter("select * from leave", con);
    DataSet ds = new DataSet();
    ada.Fill(ds, "leave");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "leave";
    con.Close();
}
private void button2_Click(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("delete from leave where emp_id="" +
comboBox1.Text + " ", con);
    MessageBox.Show("deleted");
    cmd.ExecuteNonQuery();
    con.Close();
    Refresh();
}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("select * from leave where emp_id = '" +
comboBox1.SelectedItem.ToString() + "' ", con);
    SqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        comboBox1.Text = dr.GetValue(0).ToString();
    }
}

```

```
txtEmpName.Text = dr.GetValue(1).ToString();
txtEmpDep.Text = dr.GetValue(2).ToString();
txtEmpTot.Text = dr.GetValue(3).ToString();
txtEmpWork.Text = dr.GetValue(4).ToString();
txtEmpWage.Text = dr.GetValue(5).ToString();
txtEmpSick.Text = dr.GetValue(6).ToString();
txtEmpCasual.Text = dr.GetValue(7).ToString();
txtEmpNo.Text = dr.GetValue(8).ToString();
txtEmpMon.Text = dr.GetValue(9).ToString();
txtEmpExtra.Text = dr.GetValue(10).ToString();
txtEmpDed.Text = dr.GetValue(11).ToString();
}
dr.Close();
con.Close();
}
private void button5_Click(object sender, EventArgs e)
{
    comboBox1.Text = string.Empty;
    txtEmpName.Clear();
    txtEmpDep.Clear();
    txtEmpTot.Clear();
    txtEmpWork.Clear();
    txtEmpWage.Clear();
    txtEmpSick.Clear();
    txtEmpCasual.Clear();
    txtEmpNo.Clear();
    txtEmpMon.Clear();
    txtEmpExtra.Clear();
    txtEmpDed.Clear();
}

}
```

Salary and payslip form

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
namespace final
{
    public partial class salary : Form
    {
        MySqlConnection sqlConn = new MySqlConnection();
        MySqlCommand sqlCmd = new MySqlCommand();
        MySqlDataAdapter sqlDta = new MySqlDataAdapter();
        DataTable sqlDt = new DataTable();
        MySqlDataReader sqlRd;

        DataSet DS = new DataSet();

        String server = "localhost";
        String database = "employee";
        String username = "root";
        String password = "admin";
```

```

private void uploadData()
{
    sqlConn.ConnectionString = "server=" + server + ";" + "user id=" + username + ";" +
    "password=" + password + ";" + "database=" + database;

    sqlConn.Open();

    sqlCmd.Connection = sqlConn;

    sqlCmd.CommandText = "select * from employeedb";

    sqlRd = sqlCmd.ExecuteReader();

    sqlDt.Load(sqlRd);

    sqlRd.Close();

    sqlConn.Close();

    dataGridView1.DataSource = sqlDt;
}

private void refreshdb()
{
    try
    {
        sqlConn.ConnectionString = "server=" + server + ";" + "user id=" + username + ";" +
        "password=" + password + ";" + "database=" + database;

        sqlCmd.Connection = sqlConn;

        MySqlDataAdapter sqlDta = new MySqlDataAdapter("select * from employeedb",
        sqlConn);

        DataTable sqlDt = new DataTable();

        sqlDta.Fill(sqlDt);

        dataGridView1.DataSource = sqlDt;
    }
}

```

```
        catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
}

public salary()
{
    InitializeComponent();
    uploadData();
}

private void label3_Click(object sender, EventArgs e)
{
}

private void textBox22_TextChanged(object sender, EventArgs e)
{
}

private void txtotherpayment_MouseClick(object sender, MouseEventArgs e)
{
    txtotherpayment.Text = "";
    txtotherpayment.Focus();
}

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

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

```
{  
    txtEmpID.Text = "56";  
    txtFirstname.Text = "asda";  
    txtSurname.Text = "gngdh";  
    txtaddress.Text = "gsgs";  
    txtmobile.Text = "3434";  
    txtcityweighting.Text = "123";  
    txtbasicsalary.Text = "443";  
    txtovertime.Text = "355";  
    txtotherpayment.Text = "33";  
    txttax.Text = "";  
    txtpension.Text = "";  
    txtstudentloan.Text = "";  
    txtNIPayment.Text = "";  
    txttaxcode.Text = "gsgsd";  
    txttaxperiod.Text = "2";  
    txtNIPeriod.Text = "435342";  
    txtNICode.Text = "fgsg";  
    txttaxablepay.Text = "";  
    txtpensionablepay.Text = "";  
    txtnetpay.Text = "";  
    txtgrosspay.Text = "";  
    txtdeduction.Text = "";  
    txtsearch.Text = "";  
    dateTimePicker1.ResetText();  
}  
catch (Exception ex)
```

```

    }

    MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
MessageBoxIcon.Information);

}

private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)

{
try

{
    System.Drawing.Font fntsring = new System.Drawing.Font("Arial", 18,
FontStyle.Regular);

    e.Graphics.DrawString(rtPayslip.Text, fntsring, Brushes.Black, 120, 120);

}
catch (Exception ex)

{
    MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}

}

private void btnprint_Click(object sender, EventArgs e)

{
try

{
    printPreviewDialog1.ShowDialog();
}

catch (Exception ex)

{

```

```

        MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
MessageBoxIcon.Information);

    }

}

private void btnexit_Click(object sender, EventArgs e)

{
try
{
    DialogResult iExit;

    iExit = MessageBox.Show("confirm if you want to exit", "Employee system",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);

    if (iExit == DialogResult.Yes)

    {
        Application.Exit();
    }
}
catch (Exception ex)

{
    MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}

private void salary_Load(object sender, EventArgs e)

{

}

private void btnrefresh_Click(object sender, EventArgs e)

{
    refreshdb();
}

```

```

}

private void btnupdate_Click(object sender, EventArgs e)
{
    try
    {
        sqlConn.ConnectionString = "server=" + server + ";" + "user id=" + username + ";" +
        + "password=" + password + ";" + "database=" + database;

        sqlCmd.Connection = sqlConn;

        String EmpID = txtEmpID.Text;

        String Firstname = txtFirstname.Text;

        String Surname = txtSurname.Text;

        String Address = txtaddress.Text;

        String Gender = combogender.Text;

        String Mobile = txtmobile.Text;

        String Cityweighting = txtcityweighting.Text;

        String Salary = txtbasicsalary.Text;

        String Overtime = txtovertime.Text;

        String Otherpayment = txtotherpayment.Text;

        String Tax = txttax.Text;

        String Pension = txtpension.Text;

        String Studentloan = txtstudentloan.Text;

        String NIPayment = txtNIPayment.Text;

        String Paydate = dateTimePicker1.Text;

        String Taxperiod = txttaxperiod.Text;

        String Taxcode = txttaxcode.Text;

        String NIPeriod = txtNIPeriod.Text;

        String NICode = txtNICode.Text;

        String Taxablepay = txttaxablepay.Text;
    }
}

```

```

        String Pensionablepay = txtpensionablepay.Text;

        String Netpay = txtnetpay.Text;

        String Grosspay = txtgrosspay.Text;

        String Deduction = txtdeduction.Text;

        sqlCmd.CommandText = "update from employeedb set EmpID = '" + EmpID +
        "',Firstname = '" + Firstname + "',Surname = '" + Surname + "','" +
        "Address = '" + Address + "',Gender = '" + Gender + "',Mobile = '" + Mobile +
        "',Cityweighting = '" + Cityweighting + "',Salary = '" + Salary +
        "',Overtime = '" + Overtime + "', Otherpayment = '" + Otherpayment + "',Tax = '" +
        + Tax + "',Pension = '" + Pension + "', Studentloan = '" + Studentloan +
        "',NIPayment = '" + NIPayment + "',Paydate = '" + Paydate + "',Taxperiod = '" +
        Taxperiod + "',Taxcode = '" + Taxcode + "',NIPeriod = '" + NIPeriod +
        "',NICode = '" + NICode + "',Taxablepay = '" + Taxablepay + "',Pensionablepay =
        "' + Pensionablepay + "',Netpay = '" + Netpay +
        "',Grosspay = '" + Grosspay + "',Deduction = '" + Deduction + "' WHERE ID = "
        + EmpID + "';

        Console.WriteLine("PoojaPagal " + sqlCmd.CommandText);

        sqlConn.Open();

        sqlRd = sqlCmd.ExecuteReader();

        MessageBox.Show("Record Updated", "Employee System",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

        sqlConn.Close();

        refreshdb();

    }

    catch (Exception ex)

    {

        MessageBox.Show(ex.Message);

    }

private void btndelete_Click(object sender, EventArgs e)

```

```

{
    try
    {
        sqlConn.ConnectionString = "server=" + server + ";" + "user id=" + username + ";" +
        + "password=" + password + ";" + "database=" + database;

        sqlCmd.Connection = sqlConn;

        String ID = txtEmpID.Text;

        sqlCmd.CommandText = "delete * from employeedb where ID = " + ID + "";

        sqlConn.Open();

        sqlRd = sqlCmd.ExecuteReader();

        MessageBox.Show("Record deleted", "Employee System",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

        sqlConn.Close();

        refreshdb();

    }

    catch (Exception ex)

    {
        MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
}

private void txtsearch_KeyPress(object sender, KeyPressEventArgs e)

{
    try
    {
        if (e.KeyChar == (Char)13)

        {
            DataView dv = sqlDt.DefaultView;

```

```

        dv.RowFilter = string.Format("Firstname like '%{0}%", txtsearch.Text);

        dataGridView1.DataSource = dv.ToTable();

    }

}

catch (Exception ex)

{

    MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
MessageBoxIcon.Information);

}

private void btnsearch_Click(object sender, EventArgs e)

{

    try

    {

        DataView dv = sqlDt.DefaultView;

        dv.RowFilter = string.Format("Firstname like'%{0}%", txtsearch.Text);

        dataGridView1.DataSource = dv.ToTable();

    }

    catch (Exception ex)

    {

        MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
MessageBoxIcon.Information);

    }

}

private void btntotal_Click(object sender, EventArgs e)

{

    try

    {

```

```

        double Grosspay, Deduction, Netpay, Period;

        double Intercity, Basicpay, Overtime, OtherPayment, Tax, Pension, StudentLoan,
NIPayment;

        Intercity = Double.Parse(txtcityweighting.Text);

        Basicpay = Double.Parse(txtbasicsalary.Text);

        Overtime = Double.Parse(txtovertime.Text);

        OtherPayment = Double.Parse(txtotherpayment.Text);

        Grosspay = Intercity + Basicpay + Overtime + OtherPayment;

        txtgrosspay.Text = String.Format("{0:c2}", Grosspay);

        Tax = (Grosspay * 9) / 100;

        Pension = (Grosspay * 12) / 100;

        StudentLoan = (Grosspay * 5) / 100;

        NIPayment = (Grosspay * 3) / 100;

        txttax.Text = String.Format("{0:c2}", Tax);

        txtpension.Text = String.Format("{0:c2}", Pension);

        txtstudentloan.Text = String.Format("{0:c2}", StudentLoan);

        txtNIPayment.Text = String.Format("{0:c2}", NIPayment);

        Deduction = Tax + Pension + StudentLoan + NIPayment;

        txtdeduction.Text = String.Format("{0:c2}", Deduction);

        Netpay = Grosspay - Deduction;

        txtnetpay.Text = String.Format("{0:c2}", Netpay);

        Period = Double.Parse(txttaxperiod.Text);

        txttaxablepay.Text = String.Format("{0:c2}", Period);

        txtpensionablepay.Text = String.Format("{0:c2}", Period * Pension);

        //

        sqlConn.ConnectionString = "server=" + server + ";" + "user id=" + username + ";" +
+ "password=" + password + ";" + "database=" + database;

        sqlConn.Open();

```

```

sqlCmd.Connection = sqlConn;

DateTime DateVal = DateTime.Now;

String dateInSqlFormat = string.Format("{0:yyyy-MM-dd}", DateVal);

Console.WriteLine("Pooja " + dateInSqlFormat);

sqlCmd.CommandText = "insert into
employeeedb(EmpID,Firstname,Surname,Address,Gender,Mobile,Cityweighting,"

+
"Basicsalary,Overtime,Otherpayment,Tax,Pension,Studentloan,NIPayment,Taxperiod,Taxco
de,NIPeriod,NICode,"

+ "Taxablepay,Pensionablepay,Netpay,Grosspay,Deduction,Paydate)" +

"values('" + txtEmpID.Text + "','" + txtFirstname.Text + "','" + txtSurname.Text + '
','" + txtaddress.Text + "','" +

"'" + combogender.Text + "','" + txtmobile.Text + "','" + txtcityweighting.Text + "','" +
+ txtbasicsalary.Text + "','" +

"'" + txtovertime.Text + "','" + txtotherpayment.Text + "','" + txttax.Text + "','" +
txtension.Text + "','" + txtstudentloan.Text + "','" +

txtNIPayment.Text + "','" + txttaxperiod.Text + "','" + txttaxcode.Text + "','" +
txtNIPeriod.Text + "','" +

txtNICode.Text + "','" + txttaxablepay.Text + "','" + txtpensionablepay.Text + "','" +
txtnetpay.Text + "','" + txtgrosspay.Text + "','" +

txtdeduction.Text + "','" + dateInSqlFormat + "')";

Console.WriteLine("This is C# " + sqlCmd.CommandText);

sqlCmd.ExecuteNonQuery();

sqlConn.Close();

refreshdb();

}

catch (Exception ex)

{

    MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
MessageBoxIcon.Information);

}

```

```

}

private void txtotherpayment_MouseLeave(object sender, EventArgs e)
{
    if (txtotherpayment.Text == "")
    {
        txtotherpayment.Text = "0.0";
    }
}

private void btnpayslip_Click(object sender, EventArgs e)
{
    rtPayslip.Clear();
    rtPayslip.AppendText("\t\t" + "Payslip" + "\t\t" + "\n");
    rtPayslip.AppendText("\t\t" + "-----" + "\t\t" + "\n");
    rtPayslip.AppendText("\t\t" + "EmpID" + "\t\t" + txtEmpID.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Firstname" + "\t\t" + txtFirstname.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Surname" + "\t\t" + txtSurname.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Address" + "\t\t" + txtaddress.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Gender" + "\t\t" + combogender.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Mobile" + "\t\t" + txtmobile.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Cityweight" + "\t\t" + txtcityweighting.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Salary" + "\t\t" + txtbasicsalary.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Overtime" + "\t\t" + txtovertime.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Otherpayment" + "\t\t" + txtotherpayment.Text +
"\n");
    rtPayslip.AppendText("\t\t" + "Tax" + "\t\t" + txttax.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Pension" + "\t\t" + txtEmpID.Text + "\n");
    rtPayslip.AppendText("\t\t" + "Studentloan" + "\t\t" + txtEmpID.Text + "\n");
    rtPayslip.AppendText("\t\t" + "NIPayment" + "\t\t" + txtEmpID.Text + "\n");
}

```

```

rtPayslip.AppendText("\t\t" + "Paydate" + "\t\t" + txtEmpID.Text + "\n");
rtPayslip.AppendText("\t\t" + "Taxperiod" + "\t\t" + txtEmpID.Text + "\n");
rtPayslip.AppendText("\t\t" + "Taxcode" + "\t\t" + txtEmpID.Text + "\n");
rtPayslip.AppendText("\t\t" + "NIPeriod" + "\t\t" + txtEmpID.Text + "\n");
rtPayslip.AppendText("\t\t" + "NICode" + "\t\t" + txtEmpID.Text + "\n");
rtPayslip.AppendText("\t\t" + "Taxablepay" + "\t\t" + txtEmpID.Text + "\n");
rtPayslip.AppendText("\t\t" + "Pensionablepay" + "\t\t" + txtEmpID.Text + "\n");
rtPayslip.AppendText("\t\t" + "Netpay" + "\t\t" + txtEmpID.Text + "\n");
rtPayslip.AppendText("\t\t" + "Grosspay" + "\t\t" + txtEmpID.Text + "\n");
rtPayslip.AppendText("\t\t" + "Deduction" + "\t\t" + txtEmpID.Text + "\n");
}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        txtEmpID.Text = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();
        txtFirstname.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();
        txtSurname.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();
        txtaddress.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();
        combogender.Text = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();
        txtmobile.Text = dataGridView1.SelectedRows[0].Cells[5].Value.ToString();
        txtcityweighting.Text =
            dataGridView1.SelectedRows[0].Cells[6].Value.ToString();
        txtbasicsalary.Text = dataGridView1.SelectedRows[0].Cells[7].Value.ToString();
        txtovertime.Text = dataGridView1.SelectedRows[0].Cells[8].Value.ToString();
        txtotherpayment.Text =
            dataGridView1.SelectedRows[0].Cells[9].Value.ToString();
        txttax.Text = dataGridView1.SelectedRows[0].Cells[10].Value.ToString();
    }
}

```

```

txtension.Text = dataGridView1.SelectedRows[0].Cells[11].Value.ToString();

txtstudentloan.Text = dataGridView1.SelectedRows[0].Cells[12].Value.ToString();

txtNIPayment.Text = dataGridView1.SelectedRows[0].Cells[13].Value.ToString();

dateTimePicker1.Text =
dataGridView1.SelectedRows[0].Cells[14].Value.ToString();

txttaxperiod.Text = dataGridView1.SelectedRows[0].Cells[15].Value.ToString();

txttaxcode.Text = dataGridView1.SelectedRows[0].Cells[16].Value.ToString();

txtNIPeriod.Text = dataGridView1.SelectedRows[0].Cells[17].Value.ToString();

txtNICode.Text = dataGridView1.SelectedRows[0].Cells[18].Value.ToString();

txttaxablepay.Text = dataGridView1.SelectedRows[0].Cells[19].Value.ToString();

txtpensionablepay.Text =
dataGridView1.SelectedRows[0].Cells[20].Value.ToString();

txtnetpay.Text = dataGridView1.SelectedRows[0].Cells[21].Value.ToString();

txtgrosspay.Text = dataGridView1.SelectedRows[0].Cells[22].Value.ToString();

txtdeduction.Text = dataGridView1.SelectedRows[0].Cells[23].Value.ToString();

}

catch (Exception ex)

{

    MessageBox.Show(ex.Message, "Employee System", MessageBoxButtons.OK,
MessageBoxIcon.Information);

}

}

}

}

```

System Testing

SYSTEM TESTING of software or hardware is testing on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of inner design of the code or logic. System testing includes testing for bugs in the projects. Testing is useful to check syntax and logical errors. All texts should be traceable to customer requirements. Test should be planned long before testing begins 80 percent of all errors uncovered during testing will likely be traceable to 20 percent of all program components. The testing process begins with levels of testing followed by test plan and procedures. In further steps test case is analyzed and executed. During this process if errors are uncovered then testing is rolled back to test plan and all other steps are repeated. Otherwise, if there are no errors, testing is successful and finally the test project is given.

System testing is performed on the entire system in the context of a Functional Requirement Specifications (FRS) and/or a System Requirement Specification (SRS), System testing tests only the design, but also the behaviour and even the believed expectations of the customer. It also intended to test up to and beyond the bounds defined in the software/hardware requirements specifications.

WHITE BOX TESTING

It is also called as glass-box testing. It is a test case design method that uses structure of the procedural design to derive test cases. Using this method the software engineer can derive test cases.

- 1) Guarantee that all independent paths within a module have been exercised at least once.
- 2) Exercise all logical decisions on their true and false sides.
- 3) Exercise all loops at their boundaries and within their operational bounds.
- 4) Exercise internal data structures to ensure their validity.

White-box testing of software is predicated on close examination of procedural all, the system was tested for the calculation matters were the data provided for vying the right output or not. If wrong data was provided then what it is throwing error or accepting.

BLACK BOX TESTING

It is also called as behavioural testing which focuses on the functional requirement of the software. That means it enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements of the program. Black box testing is a complimentary approach that is likely to uncover a different class of errors than white box testing. This testing is used to demonstrate that software functions are operational. That is, it ensures that input is properly accepted and output is correctly produced. The integrity of external information is maintained. It examines some fundamental aspects of a system. Black box testing attempts to find the following errors:

Incorrect or missing functions
Interface errors
Errors in external database access
Behaviour or performance errors
Initialization and termination errors

By applying black-box techniques, we derive a set of test cases that satisfy the following criteria:

- a) Test cases that reduce, by a count that is greater than one, the number of Additional test cases that must be designed to achieve reasonable testing.
- b) Test cases that tell us something about the presence or absence of classes of errors, rather than an error associated only with specific test at hand.

The attributes of white box and black box testing can be combined to provide an approach that validates the software interface and selectively ensures that internal working of software is correct. Black box testing for this system was done to check internal testing to check whether the system is working properly in each case or o and what kind of errors are there in database designs.

Future Enhancements

The future of Payroll Management system

The scope of the project includes all the future enhancements which can be done to make it more feasible.

Features which can be included for future enhancements of this project are:

This system is very flexible so that the maintenance and further amendments based on the changing environment and requirements can be made easily. Any changes that may lead to system failures are prevented with security measures. The project will support a multi-user environment, which is more than one user can access simultaneously.

It can be further developed to include more operations and analysis, as changes are required in the system to adapt to the external developments. Further enhancements can be made to the system at any later point in time.

Coding procedures can be modified according to the needs of the user. The system code is also well designed that it will form the basis for further enhancement and also new operations can be included in the system. The reports can be represented in all necessary protection. Added options can be designed in report.