

# EMPLOYEE MANAGEMENT SYSTEM

Submitted for partial fulfillment of the requirements for the

## BACHELOR OF COMPUTER APPLICATION ( BCA )

BY

Priyam Sharma

Vaishnavi Chauhan

Varsha Kumari

**Under the guidance of**

Faculty name

Assistant professor

Department of BCA and MCA



**IIMT GROUP OF COLLEGES**

# CERTIFICATE

This is to certify that the project work entitled EMPLOYEE MANAGEMENT SYSTEM is a Bonafide work carried out by <Mr. / Ms. Name of the Student (Roll No.)> in partial fulfillment of the requirements for the award of degree of BACHELOR OF COMPUTER APPLICATION by CCS UNIVERSITY, Meerut, under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Name of Guide

<Designation>

## Student's Declaration

We hereby declare that the work being presented in this report entitled "EMPLOYEE MANAGEMENT SYSTEM" is an authentic record of our own work carried out under the supervision of Dr. /Mr. /Ms. .... The matter embodied in this report has not been submitted by us for the award of any other degree.

Dated :

Signature of students(s)  
Name of student  
Department:

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

Signature of HOD  
(Name)

(Name of Department)  
Date.....

Signature of Supervisor  
(Name.....)

(Designation)  
(Name of Department)

## ACKNOWLEDGEMENT

I would like to express my sincerest gratitude and indebtedness to the person who gave me a moral and technical support & whose kind assistance has been instrumental in completion to this industrial training. It gives me immense pleasure to own me humble gratefulness to my faculty ----- for this indispensable guidance and providing necessary ideas and facilities to carry out this project.

I would like to place on record my best regards and deepest sense of gratitude to ----- (Department Head), ----- (Project Guide), of college name for their careful and precious guidance which was extremely valuable for my study both theoretically and practically.

Signature of Student  
(Name of Student in Capital)

**TABLE OF CONTENTS**

S. No.	Topic	Page No.
1	Certification	2
2	Declaration	3
3	Acknowledgement	4
4	Abstract	7
5	Chapter-1: Introduction	7
6	Chapter-2: Objective	8
7	Chapter-3: Methodology	8
8	Chapter-4: Feasibility Study	10
9	Chapter-5: Implementation and Results	11
10	Chapter-6: Entity - Relationship Diagram	51
11	Chapter-7: Data Flow Diagram	52
12	Chapter-8: Testing	54
13	Chapter-9: Conclusion	55
14	Chapter-10: Future Scope	56
15	Chapter-11: References	57

## **TABLE OF FIGURES**

<b>Figure No.</b>	<b>Caption of the Figure</b>	<b>Page No.</b>
1.	Front page	11
2.	Login frame	13
3.	Main Dashboard	16
4.	IT Department	19
5.	Finance Department	19
6.	Production Department	20
7.	Sales and marketing Department	20
8.	SQL Queries	23
9.	Tables in DB	23
10.	Add Employee	25
11.	Successfully Added message	25
12.	View Employee	32
13.	Update	37
14.	Successfully Updated message	37
15.	Remove Employee	44
16.	Successfully Removed message	44

## Abstract

*Human resource difficulties face all businesses, large and small. Because every organization has different staff management needs, we create custom employee management solutions that are tailored to your needs. This is intended to aid strategic planning and guarantee that your firm has the appropriate degree of human resources to meet your long-term objectives. This approach will help you to better manage your resources in the long run.*

### I. Introduction

Everything has been digitalized in our age of ever-increasing technology. The human workforce has grown as a result of the abundance of job options. As a result, a system that can handle the data of such a vast number of people in a company is required. Because of its user-friendly design, this project makes the process of keeping records easier. The "EMPLOYEE MANAGEMENT SYSTEM" was created to address the issues that plagued the previous manual system. This program is designed to eliminate, and in some cases, decrease, the problems that the current system has.

To eliminate data entry mistakes, the software is kept as simple as possible. When inputting incorrect data, it also displays an error notice. The user doesn't require any formal expertise to operate this system. The admin will be able to add new employees to this project. Employee data may also be seen and printed by the administrator. Admins can also remove an employee and change their details.

## **II. Objective**

The objective of this work is to give a complete approach to personnel information management. This will be accomplished by developing and deploying an HR management system that will result in a significant shift in the way employee data is managed.

This system's objectives include the following:

1. Design of an HR management system to meet needs such as adding and deleting employees, viewing, and printing employee data, and updating employee information.
2. Employee data is stored in a well-designed database.
3. An easy-to-use interface that will let user interact with the system.

## **III. Methodology**

The methodology to complete this project is as follows:

1. I explored net beans, concepts of swings and applets.
2. For further and a deeper understanding, I even referred to some articles, books, journals, websites and news articles.

Below are the important concepts on which the work has been done and with the support of these I was able to work on my project.

**NET BEANS-** NetBeans is a Java-based integrated development environment (IDE). NetBeans enables the creation of applications using a set of modular software components known as modules. NetBeans is compatible with

Windows, Mac OS X, Linux, and Solaris.

It also allows other programming languages to be extended. In addition to Java programming, Third-party developers can expand NetBeans-based applications, including the NetBeans IDE.

**JAVA-** High-level, Object-Oriented programming language which help programmers to run their applications efficiently. JAVA is the programming language which comes into our minds when we talk about android application. By using JAVA as a programming language, programmer can develop any type of android application easily.

JAVA also provides many libraries which also helps in making efficient android application. Swing is a Java GUI widget toolkit. It's part of Oracle's Java Foundation Classes (JFC), which provides an API for creating graphical-user-interfaces for Java programmers.

**SWING-** Swing is a Java GUI widget toolkit. It's part of Oracle's Java Foundation Classes (JFC), which provides an API for creating-graphical-user-interfaces for Java programmers. Swing was created to give a more advanced collection of graphical user interface components than the previous Abstract Window Toolkit (AWT).

Swing offers a pluggable look and feel that allows applications to have a look & feel that is unconnected to the underlying platform, as well as a look & feel that emulates the look & feel of numerous platforms.

**SQL-** SQL (Structured Query Language) is a computer language that is used to manage data in a relational database management system (RDBMS) or for stream processing in a relational data stream management system (RDSMS). It's especially

beneficial for dealing with structured data, or data that has relationships between entities and variables.

## **IV. Feasibility Study**

In order to do a feasibility study, we must consider the following:

### **1. Technical Feasibility**

The availability of hardware & Software necessary for the creation of the system, as-well-as the compatibility and maturity of the technology planned to be used, and the availability of the requisite technical staff to create the system, are all factors to consider.

### **2. Operational Feasibility**

Problems that may develop during operations are the focus of operation feasibility. There are two parts to this problem to consider:

- What are the chances that the solution provided will not be used or will not work?
- What is the inclination of the management and end users towards the solution? Though there is very least possibility of management being averse to the solution, there is a significant probability that the end users may not be interested in using the solution due to lack of training, insight etc.

### **3. Economic Feasibility**

The concept of economic feasibility is determining whether or not the potential benefit of fixing difficulties is worthwhile. Because member needs & alternative solutions haven't been specified at this point, it is difficult to estimate the cost at this level

It is the measure of cost effectiveness of the project. The economic feasibility is nothing but judging whether the possible benefit of solving the problems is

worthwhile or not.

## V. IMPLEMENTATION AND RESULT

Following are the screens of the Employee Management System where you can see all the features of this system in use and you can also see the GUI of the system:

**1. Front page** — This is the Entry point of our Software which will appear when we open the application. Inside this Window, the main title of the project is splashing on the top with a duration of 200ms and at the bottom, A button is placed named “ Click here to Continue ” with respect to its background image.



Fig 1

```
package employee.management.system;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class Splash extends JFrame implements ActionListener{
```

```
@Override
public void actionPerformed(ActionEvent e) {
    setVisible(false);
    new Login();
}

Splash(){
    getContentPane().setBackground(Color.WHITE);
    setLayout(null);

    JLabel heading =new JLabel("EMPLOYEE MANAGEMENT SYSTEM");
    heading.setBounds(80, 30, 1200,60);
    heading.setFont(new Font("serif",Font.PLAIN,60));
    heading.setForeground(Color.RED);
    add(heading);

    ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/front.jpg"));
    Image i2=i1.getImage().getScaledInstance(1100, 700, Image.SCALE_DEFAULT);
    ImageIcon i3= new ImageIcon(i2);
    JLabel image=new JLabel(i3);
    image.setBounds(50, 100, 1080, 500);
    add(image);

    JButton clickHere=new JButton("Click Here To Continue");
    clickHere.setBounds(400, 400, 300, 70);
    clickHere.setBackground(Color.BLACK);
    clickHere.setForeground(Color.WHITE);
    clickHere.addActionListener(this);
    image.add(clickHere);

    setVisible(true);
    setSize(1170, 650);
    setLocation(200,50);
```

```
while(true){  
    heading.setVisible(false);  
  
    try {  
        Thread.sleep(400);  
  
    } catch (Exception e) {  
  
    }  
  
    heading.setVisible(true);  
  
    try {  
        Thread.sleep(400);  
  
    } catch (Exception e) {  
  
    }  
  
}  
  
public static void main(String[] args) {  
  
    new Splash();  
}
```

**2.Login frame** – This is the login frame of this system where user have to enter the required Credentials to have access for the main dashboard.

The default Username is “admin” and Password is “123456”.

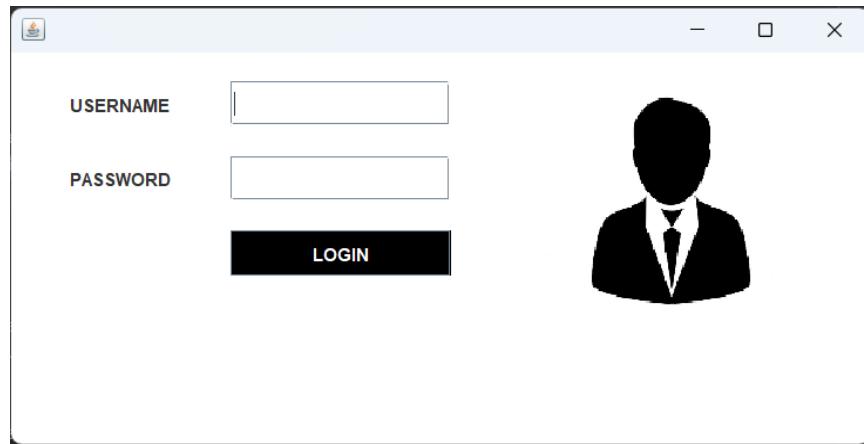


Fig 2

```

package employee.management.system;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class Login extends JFrame implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent ae) {
        try {
            String username=userText.getText();
            String password=passText.getText();
            Conn c= new Conn();

            String query ="select * from login where username = '"+username+"'and password ='"+password+"'";
            ResultSet rs= c.s.executeQuery(query);
            if(rs.next()){
                setVisible(false);
                new Home1();
            }
        }
    }
}

```

```
else{
    JOptionPane.showMessageDialog(null, "invalid Username and Password");
}

} catch (Exception e) {
    e.printStackTrace();
}

}

JTextField userText;
JTextField passText;

Login(){
    getContentPane().setBackground(Color.WHITE);
    setLayout(null);

    JLabel username=new JLabel("USERNAME ");
    username.setBounds(40,20, 100, 30);
    add(username);

    userText = new JTextField();
    userText.setBounds(150,20,150,30);
    add(userText);

    JLabel password=new JLabel("PASSWORD ");
    password.setBounds(40,70, 100, 30);
    add(password);

    passText = new JTextField();
    passText.setBounds(150,70,150,30);
    add(passText);

    JButton login=new JButton("LOGIN");
    login.setBounds(150, 120, 150, 30);
    login.setBackground(Color.BLACK);
```

```

login.setForeground(Color.WHITE);

login.addActionListener(this);

add(login);

ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/second.jpg"));

Image i2=i1.getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT);

ImageIcon i3= new ImageIcon(i2);

JLabel image=new JLabel(i3);

image.setBounds(350, 0, 200, 200);

add(image);

setSize(600, 300);

setLocation(450 ,250);

setVisible(true);

}

public static void main(String[] args) {

new Login();

}

}

```

**3. Main Dashboard –** After login in, user is directed to the main dashboard of this system where you can select the department

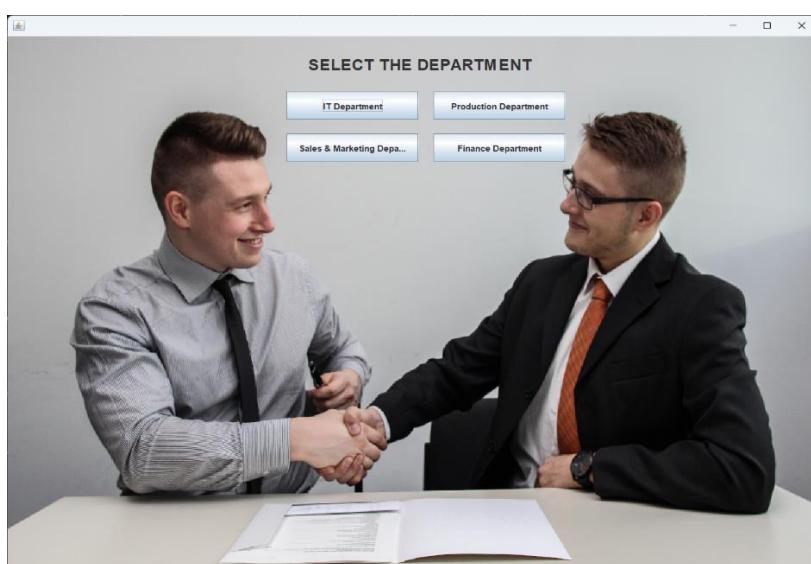


Fig 3

```

package employee.management.system;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Home1 extends JFrame implements ActionListener {
    JButton IT,prod,sales,finance;

    @Override
    public void actionPerformed(ActionEvent ae) {
        if(ae.getSource() == IT){
            setVisible(false);
            new employee.management.system.ITCrud.Home2();
        }
        else if(ae.getSource() == prod){
            setVisible(false);
            new employee.management.system.productionCrud.Home2();
        }
        else if(ae.getSource() == sales){
            setVisible(false);
            new employee.management.system.salesCrud.Home2();
        }
        else{
            setVisible(false);
            new employee.management.system.financeCrud.Home2();
        }
    }

    public Home1(){
        setLayout(null);

        ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/dept.jpg"));
        Image i2=i1.getImage().getScaledInstance(1120, 800, Image.SCALE_DEFAULT);
    }
}

```

```
ImageIcon i3= new ImageIcon(i2);
JLabel image=new JLabel(i3);
image.setBounds(0, 0, 1120, 800);
add(image);

JLabel heading =new JLabel("SELECT THE DEPARTMENT");
heading.setBounds(410, 20, 400, 40);
heading.setFont(new Font("segou UI",Font.BOLD,22));
image.add(heading);

IT= new JButton("IT Department");
IT.setBounds(380, 80 ,180, 40);
IT.addActionListener(this);
image.add(IT);

prod= new JButton("Production Department");
prod.setBounds(580, 80 ,180, 40);
prod.addActionListener(this);
image.add(prod);

sales= new JButton("Sales & Marketing Department");
sales.setBounds(380, 140 ,180, 40);
sales.addActionListener(this);
image.add(sales);

finance= new JButton("Finance Department");
finance.setBounds(580, 140 ,180, 40);
finance.addActionListener(this);
image.add(finance);

setSize(1120, 800);
 setLocation(250, 50);
setVisible(true);
```

```
}
```

```
public static void main(String[] args) {
```

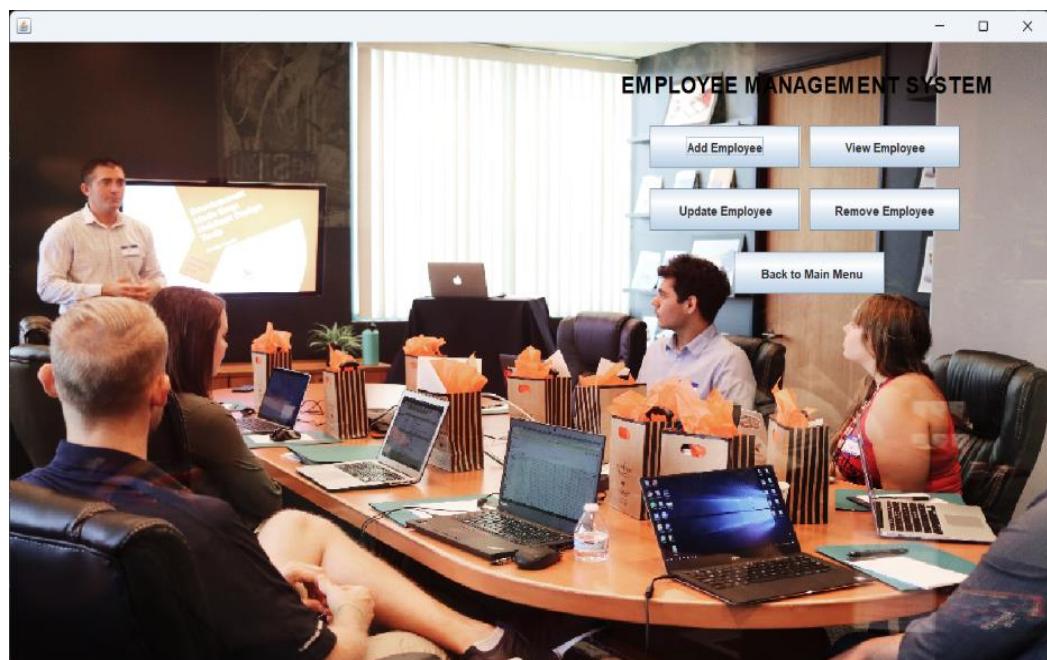
```
    new Home1();
```

```
}
```

```
}
```

**4. Interfaces of different departments:** You can add, view, remove or update employee details or can go back to Main menu.

**i) IT Department-**



*Fig 4*

**ii) Finance Department-**



Fig 5

**iii) Production Department-**

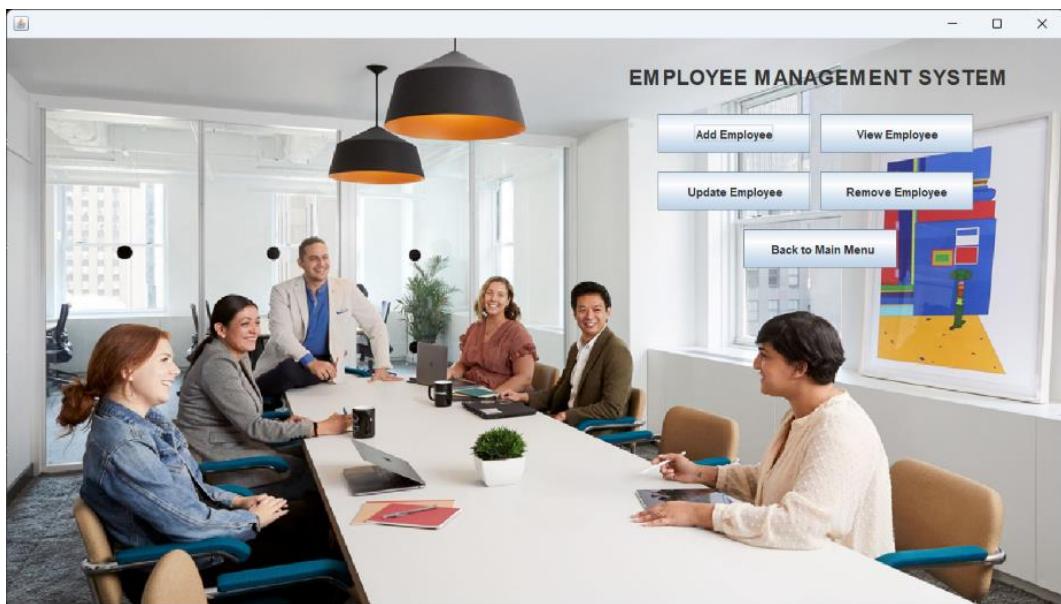


Fig 6

**iv) Sales & marketing Department-**



Fig 7

```

package employee.management.system.ITScrud;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Home2 extends JFrame implements ActionListener {
    JButton add,view,update,remove,back;

    public Home2(){
        setLayout(null);

        // below line is setting the background for every department to be selected
        ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/ITDept.jpg"));
        Image i2=i1.getImage().getScaledInstance(1120, 630, Image.SCALE_DEFAULT);
        ImageIcon i3= new ImageIcon(i2);
        JLabel image=new JLabel(i3);
        image.setBounds(0, 0, 1120, 630);
        add(image);

        JLabel heading =new JLabel("EMPLOYEE MANAGEMENT SYSTEM");
        heading.setBounds(650, 20, 400, 40);
        heading.setFont(new Font("segou UI",Font.BOLD,22));
    }
}

```

```
heading.setForeground(Color.black);

image.add(heading);

add= new JButton("Add Employee");

add.setBounds(680, 80 ,160, 40);

add.addActionListener(this);

image.add(add);

view= new JButton("View Employee");

view.setBounds(850, 80 ,160, 40);

view.addActionListener(this);

image.add(view);

update= new JButton("Update Employee");

update.setBounds(680, 140 ,160, 40);

update.addActionListener(this);

image.add(update);

remove= new JButton("Remove Employee");

remove.setBounds(850, 140 ,160, 40);

remove.addActionListener(this);

image.add(remove);

back= new JButton(" Back to Main Menu");

back.setBounds(770, 200 ,160, 40);

back.addActionListener(this);

image.add(back);

setSize(1120, 630);

setLocation(250, 100);

setVisible(true);

}
```

```

@Override
public void actionPerformed(ActionEvent ae) {
    if(ae.getSource() == add){
        setVisible(false);
        new AddEmployee();
    }
    else if(ae.getSource() == view){
        setVisible(false);
        new ViewEmployee();
    }
    else if(ae.getSource() == update){
        setVisible(false);
        new ViewEmployee();
    }
    else if (ae.getSource()==remove){
        setVisible(false);
        new RemoveEmployee();
    }
    else{
        setVisible(false);
        new employee.management.system.Home1();
    }
}

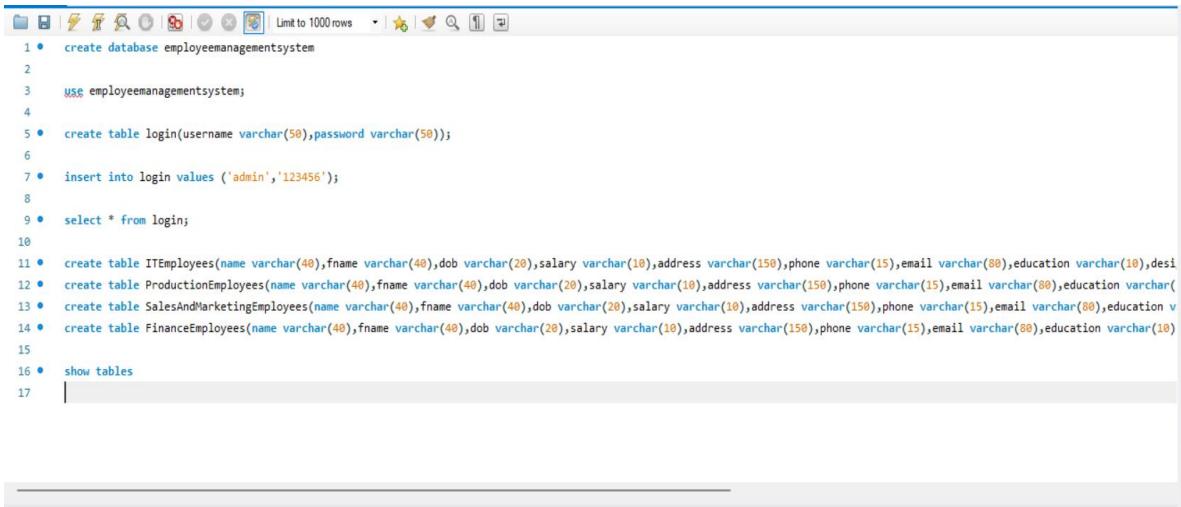
public static void main(String[] args) {
    new Home2();
}
}

```

## 5. Operations-

- **Connecting Database:** MySQL Database is used in backend for storing the data.

Below are the SQL queries used to create the database named “employeemanagementsystem” using MySQL Workbench.



```

1 •  create database employeemanagementsystem
2
3 •  use employeemanagementsystem;
4
5 •  create table login(username varchar(50),password varchar(50));
6
7 •  insert into login values ('admin','123456');
8
9 •  select * from login;
10
11 •  create table ITEmployees(name varchar(40),fname varchar(40),dob varchar(20),salary varchar(10),address varchar(150),phone varchar(15),email varchar(80),education varchar(10),desi
12 •  create table ProductionEmployees(name varchar(40),fname varchar(40),dob varchar(20),salary varchar(10),address varchar(150),phone varchar(15),email varchar(80),education varchar(
13 •  create table SalesAndMarketingEmployees(name varchar(40),fname varchar(40),dob varchar(20),salary varchar(10),address varchar(150),phone varchar(15),email varchar(80),education v
14 •  create table FinanceEmployees(name varchar(40),fname varchar(40),dob varchar(20),salary varchar(10),address varchar(150),phone varchar(15),email varchar(80),education varchar(10)
15
16 •  show tables
17

```

Fig. 8

Tables_in_employeemanagementsystem
financeemployees
itemployees
▶ login
productionemployees
salesandmarketingemployees

Fig. 9

Now, the code used to connect with the database is given below:

```

package employee.management.system;

import java.sql.*;

public class Conn {

    public Connection c;
    public Statement s;

    public Conn(){
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            c=DriverManager.getConnection("jdbc:mysql://localhost:3306/employeemanagementsystem","root", "root");
        }
    }
}

```

```
s=c.createStatement();  
  
    } catch (Exception e) {  
  
        e.printStackTrace();  
  
    }  
  
}
```

- i) **Add Employee Details:** By clicking on “Add Employee” Button on previous menu, you will redirect to another window where you can add new employees by filling details of the employee and after clicking on “Add Employee” below message will appear.10

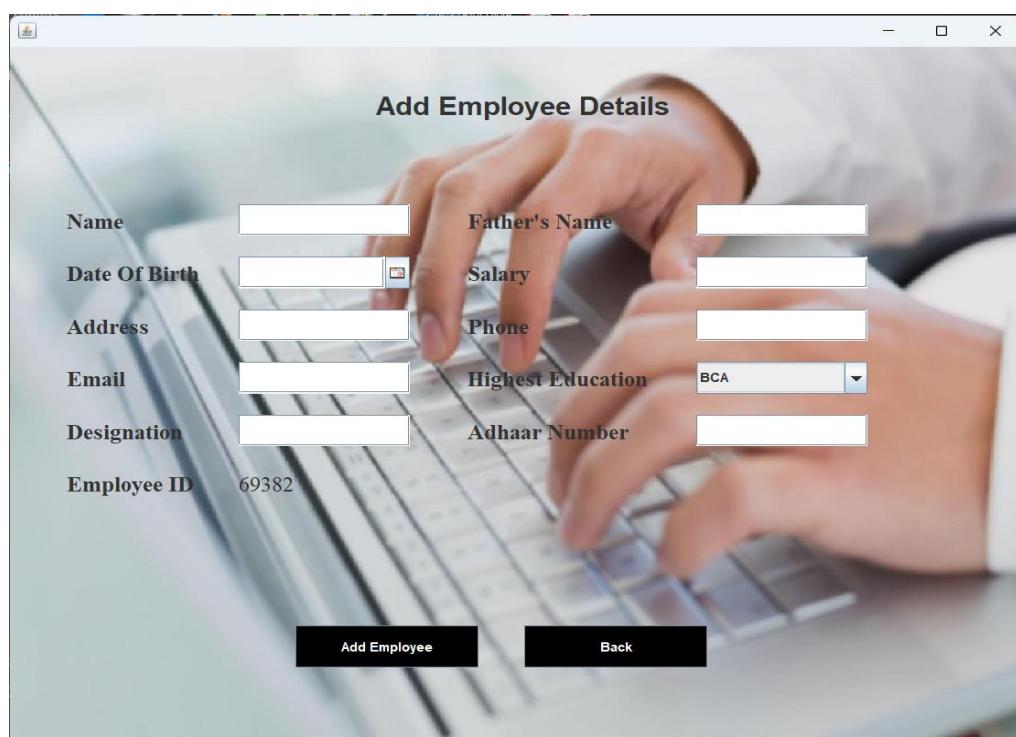


Fig 10

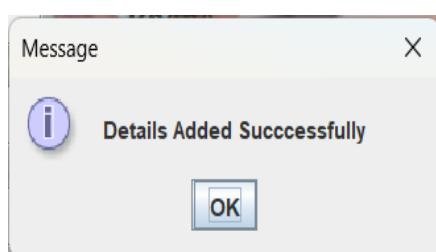


Fig 11

```

package employee.management.system.ITScrud;
import employee.management.system.*;
import java.awt.*;
import javax.swing.*;
import com.toedter.calendar.JDateChooser;
import java.awt.event.*;
import java.util.Random;

public class AddEmployee extends JFrame implements ActionListener {

    JTextField tname, tfname, tsalary, taddress, tphone, temail, tdesignation, taadhaar;
    JLabel tempid;
    JDateChooser tdob;
    JComboBox teducation;
    JButton add, back;

    @Override
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == add) {
            String name = tname.getText();
            String fname = tfname.getText();
            String dob = ((JTextField) tdob.getDateEditor().getUiComponent()).getText();
            String salary = tsalary.getText();
            String address = taddress.getText();
            String phone = tphone.getText();
            String email = temail.getText();
            String education = (String) teducation.getSelectedItem();
            String designation = tdesignation.getText();
            String aadhaar = taadhaar.getText();
            String empId = tempid.getText();

            try {
                Conn c = new Conn();

```

```

String query = "insert into ITEmployees values ('" + name + "', '" + fname + "', '" + dob + "','" + salary +
"','" + address + "','" + phone + "','" + email + "','" + education + "','" + designation + "','" + aadhaar + "','" + empld +
"');";

c.s.executeUpdate(query);

JOptionPane.showMessageDialog(null, "Details Added Successfully");

setVisible(false);

new Home2();

} catch (Exception e) {

e.printStackTrace();

}

}

}

else {

setVisible(false);

new Home2();

}

}

AddEmployee() {

Random n = new Random();

int number = n.nextInt(999999);

getContentPane().setBackground(Color.white);

setLayout(null);

//heading

JLabel heading = new JLabel("Add Employee Details");

heading.setBounds(320, 30, 500, 50);

heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));

add(heading);

//name

JLabel name = new JLabel("Name");

name.setBounds(50, 150, 150, 30);

name.setFont(new Font("SERIF", Font.BOLD, 20));

add(name);
}

```

```
tname = new JTextField();
tname.setBounds(200, 150, 150, 30);
add(tname);

//father name
JLabel fname = new JLabel("Father's Name");
fname.setBounds(400, 150, 150, 30);
fname.setFont(new Font("SERIF", Font.BOLD, 20));
add(fname);

tfname = new JTextField();
tfname.setBounds(600, 150, 150, 30);
add(tfname);

//dob
JLabel dob = new JLabel("Date Of Birth");
dob.setBounds(50, 200, 150, 30);
dob.setFont(new Font("SERIF", Font.BOLD, 20));
add(dob);

tdob = new JDateChooser();
tdob.setBounds(200, 200, 150, 30);
add(tdob);

//salary
JLabel salary = new JLabel("Salary");
salary.setBounds(400, 200, 150, 30);
salary.setFont(new Font("SERIF", Font.BOLD, 20));
add(salary);

tsalary = new JTextField();
tsalary.setBounds(600, 200, 150, 30);
add(tsalary);
```

```
//address  
  
JLabel address = new JLabel("Address");  
address.setBounds(50, 250, 150, 30);  
address.setFont(new Font("SERIF", Font.BOLD, 20));  
add(address);  
  
  
taddress = new JTextField();  
taddress.setBounds(200, 250, 150, 30);  
add(taddress);  
  
  
//phone  
  
JLabel phone = new JLabel("Phone");  
phone.setBounds(400, 250, 150, 30);  
phone.setFont(new Font("SERIF", Font.BOLD, 20));  
add(phone);  
  
  
tphone = new JTextField();  
tphone.setBounds(600, 250, 150, 30);  
add(tphone);  
  
  
// email  
  
JLabel email = new JLabel("Email");  
email.setBounds(50, 300, 150, 30);  
email.setFont(new Font("SERIF", Font.BOLD, 20));  
add(email);  
  
  
temail = new JTextField();  
temail.setBounds(200, 300, 150, 30);  
add(temail);  
  
  
//education  
  
JLabel education = new JLabel("Highest Education");  
education.setBounds(400, 300, 170, 30);
```

```
education.setFont(new Font("SERIF", Font.BOLD, 20));
add(education);

String courses[] = {"BCA", "B.Tech", "BBA", "BA", "B.Com", "B.Sc", "BJMC", "B.Ed", "MBA", "M.Tech",
"M.Sc", "PhD",};

teducation = new JComboBox(courses);
teducation.setBounds(600, 300, 150, 30);
add(teducation);

//designation
JLabel designation = new JLabel("Designation");
designation.setBounds(50, 350, 150, 30);
designation.setFont(new Font("SERIF", Font.BOLD, 20));
add(designation);

tdesignation = new JTextField();
tdesignation.setBounds(200, 350, 150, 30);
add(tdesignation);

//aadhar
JLabel aadhaar = new JLabel("Adhaar Number");
aadhaar.setBounds(400, 350, 150, 30);
aadhaar.setFont(new Font("SERIF", Font.BOLD, 20));
add(aadhaar);

taadhaar = new JTextField();
taadhaar.setBounds(600, 350, 150, 30);
add(taadhaar);

//employee id
JLabel empid = new JLabel("Employee ID");
empid.setBounds(50, 400, 150, 30);
empid.setFont(new Font("SERIF", Font.BOLD, 20));
add(empid);
```

```
tempid = new JLabel("'" + number);

tempid.setBounds(200, 400, 150, 30);

tempid.setFont(new Font("SERIF", Font.PLAIN, 20));

add(tempid);

add = new JButton("Add Employee");

add.setBounds(250, 550, 160, 40);

add.setBackground(Color.BLACK);

add.setForeground(Color.WHITE);

add.addActionListener(this);

add(add);

back = new JButton("Back");

back.setBounds(450, 550, 160, 40);

back.setBackground(Color.BLACK);

back.setForeground(Color.WHITE);

back.addActionListener(this);

add(back);

ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/add_employee.jpg"));

Image i2=i1.getImage().getScaledInstance(900, 700, Image.SCALE_SMOOTH);

ImageIcon i3= new ImageIcon(i2);

JLabel image=new JLabel(i3);

image.setBounds(0, 0, 900, 700);

add(image);

setSize(900, 700);

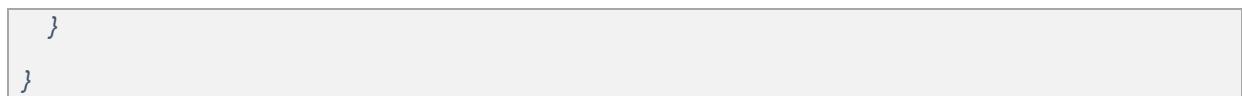
setLocation(300, 50);

setVisible(true);

}

public static void main(String[] args) {

new AddEmployee();
```



- ii) **View Employee Details:** You can view the list of employees of department selected previously in tabular format.

The screenshot shows a Windows-style application window titled "EMPLOYEE VIEW". On the left, there is a search bar labeled "Search by Employee ID" containing the value "502461". Below the search bar are four buttons: "Search", "Print", "Update", and "Back". To the right of the search bar, the word "EMPLOYEE" is written in large, bold, dark gray letters, with a magnifying glass icon integrated into the letter "O". Below "EMPLOYEE", the word "VIEW" is written in blue, bold, sans-serif letters. A horizontal line separates the search area from a table below. The table has a header row with columns for name, fname, dob, salary, address, phone, email, education, designation, aadhaar, and empid. Below the header, there are two data rows:

name	fname	dob	salary	address	phone	email	education	designation	aadhaar	empid
Shubham Sh...	Major Sharma	Nov 5, 2001	20000	block-a,sec-2...	8744833158	shubh5nov@...	B.Tech	Software Eng...	874523651254	502461
dheeraj Kuma...	Raj Kumar	Jun 14, 1996	30000	Beta 1, great...	7489564502	dheer12@hot...	BCA	Test Enginner	421595687548	105447

*Fig 13*

- **Search Employee Details:** You can Search an Employee's details by their unique Employee ID.
- **Print Employee Details:** You can Print employee details in PDF format by clicking Print using your Default Windows Printer.

- **Update Employee Details:** You can also redirect to the Update operation by selecting the Employee ID in the above drop down menu and then click on “Update” button
- **Back:** This button redirects you to previous menu.

```

package employee.management.system.ITScrud;

import employee.management.system.*;

import java.awt.*;
import java.awt.event.*;
import java.awt.print.PrinterException;
import javax.swing.*;
import net.proteanit.sql.DbUtils;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ViewEmployee extends JFrame implements ActionListener {

    JTable table;
    Choice employeedid;
    JButton Search, Print, Update, Back;

    ViewEmployee() {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel search = new JLabel("Search by Employee ID");
        search.setBounds(20, 40, 150, 20);
        add(search);

        employeedid = new Choice();
    }
}

```

```

employeeid.setBounds(180, 40, 150, 20);
add(employeeid);

try {
    Conn c = new Conn();
    ResultSet rs = c.s.executeQuery("select * from ITEmployees");
    while (rs.next()) {
        employeeid.add(rs.getString("empid"));
    }
} catch (Exception e) {
    e.printStackTrace();
}

table = new JTable();

try {
    Conn c = new Conn();
    String query = "select * from ITEmployees";
    ResultSet rs = c.s.executeQuery(query);
    table.setModel(DbUtils.resultSetToTableModel(rs));
} catch (Exception e) {
    e.printStackTrace();
}

JScrollPane jsp = new JScrollPane(table);
jsp.setBounds(0, 150, 900, 600);
add(jsp);

Search = new JButton("Search");
Search.setBounds(20, 110, 80, 20);
Search.addActionListener(this);
add(Search);

```

```

Print = new JButton("Print");
Print.setBounds(120, 110, 80, 20);
Print.addActionListener(this);
add(Print);

Update = new JButton("Update");
Update.setBounds(220, 110, 80, 20);
Update.addActionListener(this);
add(Update);

Back = new JButton("Back");
Back.setBounds(320, 110, 80, 20);
Back.addActionListener(this);
add(Back);

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/viewemp.png"));
Image i2 = i1.getImage().getScaledInstance(404, 125, Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel image = new JLabel(i3);
image.setBounds(450, 10, 404, 125);
add(image);

setSize(900, 700);
 setLocation(300, 100);
 setVisible(true);

}

@Override
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == Search) {
        String query = "select * from ITEmployees where empid = " + employeeid.getSelectedItem() + "";
}

```

```

try {
    Conn c = new Conn();
    ResultSet rs = c.s.executeQuery(query);
    table.setModel(DbUtils.resultSetToTableModel(rs));
} catch (Exception e) {
    e.printStackTrace();
}

} else if (ae.getSource() == Print) {

try {
    table.print();
} catch (PrinterException ex) {
    Logger.getLogger(ViewEmployee.class.getName()).log(Level.SEVERE, null, ex);
}

} else if (ae.getSource() == Update) {

setVisible(false);
new UpdateEmployee(employeeid.getSelectedItem());
} else {

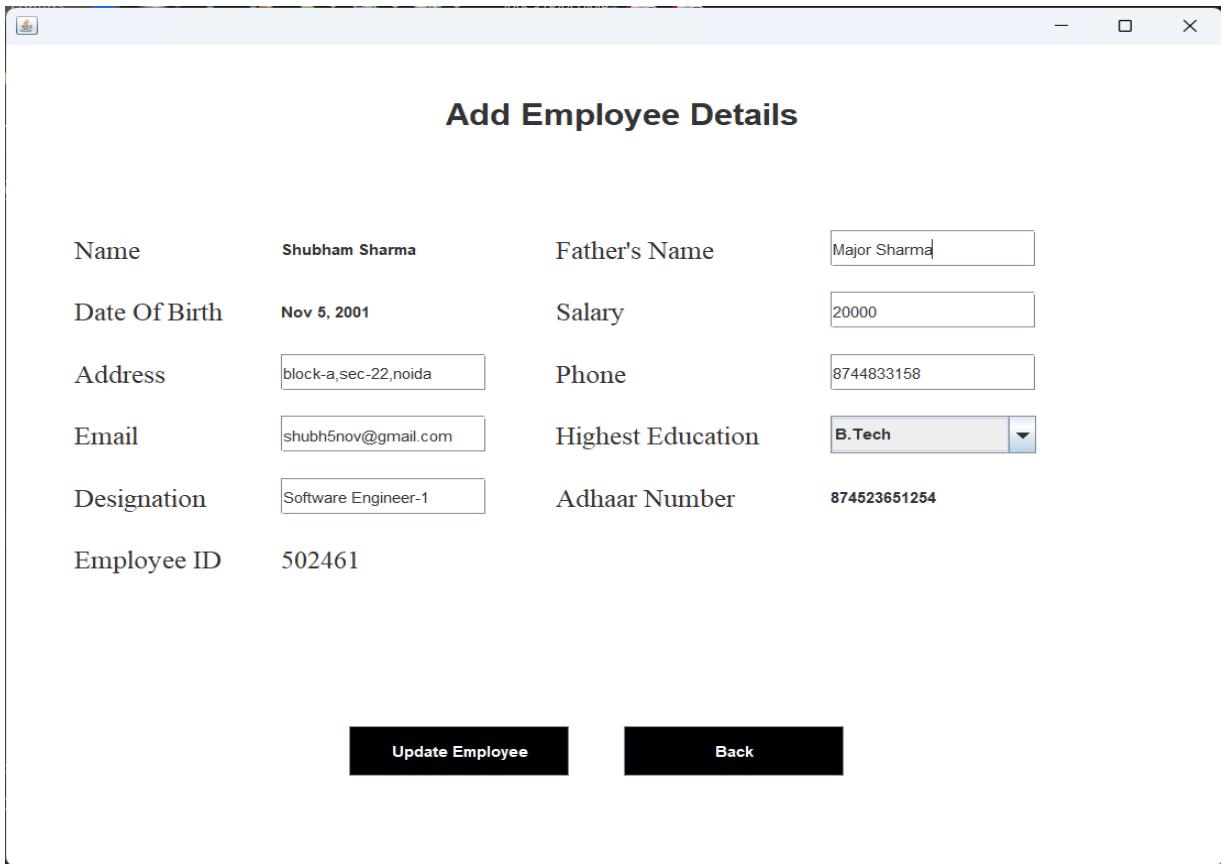
setVisible(false);
new Home2();
}
}

public static void main(String[] args) {
    new ViewEmployee();
}
}

```

- iii) **Update Employee Details:** You can make updates in employee details using

employee id by Simply clicking on “Update Employee” button . and then you will redirect to “View Employee page ” from where you can search by Employee ID and click on “Update” button .This will show you the window which has Existing details of an Employee and Now you can Edit the Selected Employee Details.



The screenshot shows a Windows-style application window titled "Add Employee Details". The form contains the following data:

Name	<b>Shubham Sharma</b>	Father's Name	Major Sharma
Date Of Birth	<b>Nov 5, 2001</b>	Salary	20000
Address	block-a,sec-22,noida	Phone	8744833158
Email	shubh5nov@gmail.com	Highest Education	B.Tech
Designation	Software Engineer-1	Adhaar Number	874523651254
Employee ID	502461		

At the bottom, there are two buttons: "Update Employee" and "Back".

Fig 14

After Clicking on “Update Employee” button , this message will pop up.

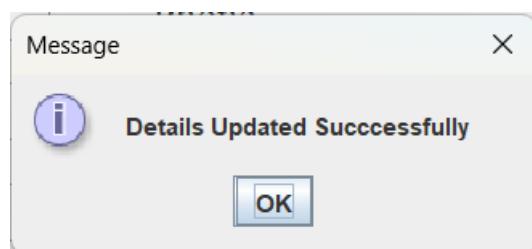


Fig 15

```
package employee.management.system.ITScrud;
import employee.management.system.*;
```

```

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;

public class UpdateEmployee extends JFrame implements ActionListener {

    JTextField fname, tsalary, taddress, tphone, temail, tdesignation; // this variables are declared as they are
    //editable

    JLabel tname, tdob, taadhaar, tempid; // these are not editable

    JComboBox teducation; // it is also an editable

    JButton update, back;

    String empid;

    UpdateEmployee(String empid) {
        this.empid = empid;
        getContentPane().setBackground(Color.white);
        setLayout(null);

        //heading
        JLabel heading = new JLabel("Add Employee Details");
        heading.setBounds(320, 30, 500, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));
        add(heading);

        //name
        JLabel name = new JLabel("Name");
        name.setBounds(50, 150, 150, 30);
        name.setFont(new Font("SERIF", Font.PLAIN, 20));
        add(name);

        tname = new JLabel();
        tname.setBounds(200, 150, 150, 30);
        add(tname);
    }
}

```

```
//father name  
  
JLabel fname = new JLabel("Father's Name");  
fname.setBounds(400, 150, 150, 30);  
fname.setFont(new Font("SERIF", Font.PLAIN, 20));  
add(fname);  
  
  
tfname = new JTextField();  
tfname.setBounds(600, 150, 150, 30);  
add(tfname);  
  
  
//dob  
  
JLabel dob = new JLabel("Date Of Birth");  
dob.setBounds(50, 200, 150, 30);  
dob.setFont(new Font("SERIF", Font.PLAIN, 20));  
add(dob);  
  
  
tdob = new JLabel();  
tdob.setBounds(200, 200, 150, 30);  
add(tdob);  
  
  
//salary  
  
JLabel salary = new JLabel("Salary");  
salary.setBounds(400, 200, 150, 30);  
salary.setFont(new Font("SERIF", Font.PLAIN, 20));  
add(salary);  
  
  
tsalary = new JTextField();  
tsalary.setBounds(600, 200, 150, 30);  
add(tsalary);  
  
  
//address  
  
JLabel address = new JLabel("Address");  
address.setBounds(50, 250, 150, 30);
```

```

address.setFont(new Font("SERIF", Font.PLAIN, 20));
add(address);

taddress = new JTextField();
taddress.setBounds(200, 250, 150, 30);
add(taddress);

//phone
JLabel phone = new JLabel("Phone");
phone.setBounds(400, 250, 150, 30);
phone.setFont(new Font("SERIF", Font.PLAIN, 20));
add(phone);

tphone = new JTextField();
tphone.setBounds(600, 250, 150, 30);
add(tphone);

// email
JLabel email = new JLabel("Email");
email.setBounds(50, 300, 150, 30);
email.setFont(new Font("SERIF", Font.PLAIN, 20));
add(email);

temail = new JTextField();
temail.setBounds(200, 300, 150, 30);
add(temail);

//education
JLabel education = new JLabel("Highest Education");
education.setBounds(400, 300, 150, 30);
education.setFont(new Font("SERIF", Font.PLAIN, 20));
add(education);

String courses[] = {"BCA", "B.Tech", "BBA", "BA", "B.Com", "B.Sc", "BJMC", "B.Ed", "MBA", "M.Tech",
" M.Sc", "PhD",};

```

```
teducation = new JComboBox(courses);
teducation.setBounds(600, 300, 150, 30);
add(teducation);

//designation
JLabel designation = new JLabel("Designation");
designation.setBounds(50, 350, 150, 30);
designation.setFont(new Font("SERIF", Font.PLAIN, 20));
add(designation);

tdesignation = new JTextField();
tdesignation.setBounds(200, 350, 150, 30);
add(tdesignation);

//aadhar
JLabel aadhaar = new JLabel("Adhaar Number");
aadhaar.setBounds(400, 350, 150, 30);
aadhaar.setFont(new Font("SERIF", Font.PLAIN, 20));
add(aadhaar);

taadhaar = new JLabel();
taadhaar.setBounds(600, 350, 150, 30);
add(taadhaar);

//employee id
JLabel empidL = new JLabel("Employee ID");
empidL.setBounds(50, 400, 150, 30);
empidL.setFont(new Font("SERIF", Font.PLAIN, 20));
add(empidL);

tempid = new JLabel();
tempid.setBounds(200, 400, 150, 30);
tempid.setFont(new Font("SERIF", Font.PLAIN, 20));
```

```

add(tempid);

try {
    //this try catch is used to parse data from database
    Conn c = new Conn();
    String query = "select * from ITEmployees where empid = '" + empid + "'";
    ResultSet rs = c.s.executeQuery(query);
    while (rs.next()) {
        // setting the data from database to these labels and textfields
        tname.setText(rs.getString("name"));
        tfname.setText(rs.getString("fname"));
        tdob.setText(rs.getString("dob"));
        tsalary.setText(rs.getString("salary"));
        taddress.setText(rs.getString("address"));
        tphone.setText(rs.getString("phone"));
        temail.setText(rs.getString("email"));
        tdesignation.setText(rs.getString("designation"));
        teducation.setSelectedItem(rs.getString("education"));
        taadhaar.setText(rs.getString("aadhaar"));
        tempid.setText(rs.getString("empid"));

    }
} catch (Exception e) {
    e.printStackTrace();
}

update = new JButton("Update Employee");
update.setBounds(250, 550, 160, 40);
update.setBackground(Color.BLACK);
update.setForeground(Color.WHITE);
update.addActionListener(this);
add(update);

back = new JButton("Back");

```

```

back.setBounds(450, 550, 160, 40);
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);
back.addActionListener(this);
add(back);

setSize(900, 700);
setLocation(300, 50);
setVisible(true);
}

@Override
public void actionPerformed(ActionEvent ae) {
if (ae.getSource() == update) {
String fname = tfname.getText();
String salary = tsalary.getText();
String address = taddress.getText();
String phone = tphone.getText();
String email = temail.getText();
String education = (String) teducation.getSelectedItem();
String designation = tdesignation.getText();

try {
// this used to update the new details into database
Conn c = new Conn();
String query ="update ITEmployees set fname = "+fname+", salary = "+salary+", address =
"+address+", phone = "+phone+", email = "+email+", education = "+education+", designation =
"+designation+" where empId = "+empid+"";
c.s.executeUpdate(query);
JOptionPane.showMessageDialog(null, "Details Updated Successfully");
setVisible(false);
new Home2();
} catch (Exception e) {
e.printStackTrace();
}
}
}

```

```

    }

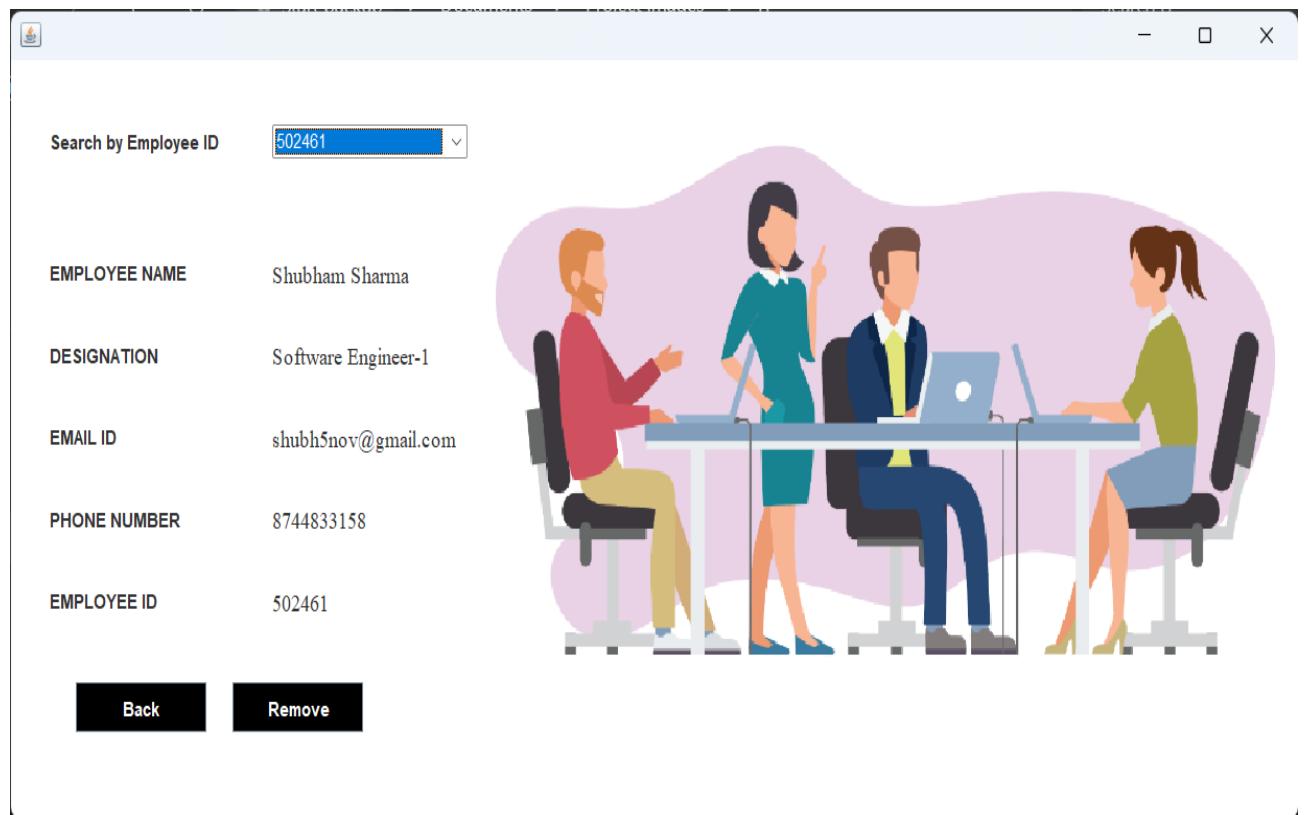
} else {
    setVisible(false);
    new Home2();
}

}

public static void main(String[] args) {
    new UpdateEmployee("");
}
}

```

- iv) **Remove Employee Details:** You can remove employee by Searching for a particular Employee ID , and their details will pop up below .



*Fig 16*

After clicking on “Remove ” Button , This message will pop up ,

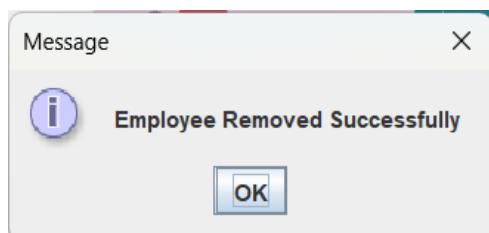


Fig 17

```

package employee.management.system.ITScrud;

import employee.management.system.*;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;

public class RemoveEmployee extends JFrame implements ActionListener {

    Choice cempid;
    JButton back, remove;

    RemoveEmployee() {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel heading = new JLabel("Search by Employee ID");
        heading.setBounds(30, 40, 150, 20);
        add(heading);

        cempid = new Choice();
        cempid.setBounds(200, 40, 150, 20);
        add(cempid);

        try {
            Conn c = new Conn();

```

```

String query = "select * from ITEmployees";
ResultSet rs = c.s.executeQuery(query);
while (rs.next()) {
    cempid.add(rs.getString("empid"));
}
} catch (Exception e) {
    e.printStackTrace();
}

JLabel name = new JLabel("EMPLOYEE NAME ");
name.setBounds(30, 120, 200, 20);
add(name);

JLabel tname = new JLabel();
tname.setBounds(200, 120, 200, 20);
tname.setFont(new Font("serif", Font.PLAIN, 15));
add(tname);

JLabel designation = new JLabel("DESIGNATION ");
designation.setBounds(30, 170, 200, 20);
add(designation);

JLabel tdesignation = new JLabel();
tdesignation.setBounds(200, 170, 200, 20);
tdesignation.setFont(new Font("serif", Font.PLAIN, 15));
add(tdesignation);

JLabel email = new JLabel("EMAIL ID ");
email.setBounds(30, 220, 200, 20);
add(email);

JLabel temail = new JLabel();
temail.setBounds(200, 220, 200, 20);
temail.setFont(new Font("serif", Font.PLAIN, 15));

```

```
add(temail);

JLabel phone = new JLabel("PHONE NUMBER ");
phone.setBounds(30, 270, 200, 20);
add(phone);

JLabel tphone = new JLabel();
tphone.setBounds(200, 270, 200, 20);
tphone.setFont(new Font("serif", Font.PLAIN, 15));
add(tphone);

JLabel empid = new JLabel("EMPLOYEE ID ");
empid.setBounds(30, 320, 200, 20);
add(empid);

JLabel tempid = new JLabel();
tempid.setBounds(200, 320, 200, 20);
tempid.setFont(new Font("serif", Font.PLAIN, 15));
add(tempid);

back = new JButton("Back");
back.setBounds(50, 380, 100, 30);
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);
back.addActionListener(this);
add(back);

remove = new JButton("Remove");
remove.setBounds(170, 380, 100, 30);
remove.setBackground(Color.BLACK);
remove.setForeground(Color.WHITE);
remove.addActionListener(this);
add(remove);
```

```

try {

    Conn c = new Conn();

    String query = "select * from ITEmployees where empid = " + cempid.getSelectedItem() + "";

    ResultSet rs = c.s.executeQuery(query);

    while (rs.next()) {

        tname.setText(rs.getString("name"));

        tdesignation.setText(rs.getString("designation"));

        tphone.setText(rs.getString("phone"));

        temail.setText(rs.getString("email"));

        tempid.setText(rs.getString("empid"));

    }

}

} catch (Exception e) {

    e.printStackTrace();

}

cempid.addItemListener(new ItemListener() {

    @Override

    public void itemStateChanged(ItemEvent ae) {

        try {

            Conn c = new Conn();

            String query = "select * from ITEmployees where empid = " + cempid.getSelectedItem() + "";

            ResultSet rs = c.s.executeQuery(query);

            while (rs.next()) {

                tname.setText(rs.getString("name"));

                tdesignation.setText(rs.getString("designation"));

                tphone.setText(rs.getString("phone"));

                temail.setText(rs.getString("email"));

                tempid.setText(rs.getString("empid"));

            }

        }

    }

}

} catch (Exception e) {

    e.printStackTrace();

}

```

```

    }

    });

    ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/delete.png"));

    Image i2 = i1.getImage().getScaledInstance(600, 400, Image.SCALE_DEFAULT);

    ImageIcon i3 = new ImageIcon(i2);

    JLabel image = new JLabel(i3);

    image.setBounds(370, 0, 600, 400);

    add(image);

    setSize(1000, 500);

    setLocation(300, 150);

    setVisible(true);

}

@Override

public void actionPerformed(ActionEvent ae) {

    if (ae.getSource() == remove) {

        try {

            Conn c = new Conn();

            String query = "delete from ITEmployees where empid = '" + cempid.getSelectedItem() + "'";

            c.s.executeUpdate(query);

            JOptionPane.showMessageDialog(null, "Employee Removed Successfully ");

            setVisible(false);

            new Home2();

        } catch (Exception e) {

            e.printStackTrace();

        }

    } else {

        setVisible(false);

        new Home2();

    }

}

```

```
public static void main(String[] args) {  
    new RemoveEmployee();  
}  
}
```

## User Interface Design:

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs.

The overall flow of screens and messages is called a dialogue.

The following steps are various guidelines for User Interface Design:

- The system user should always be aware of what to do next.
- The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
- Message, instructions or information should be displayed long enough to allow the system user to read them.
- Use display attributes sparingly.
- Default values for fields and answers to be entered by the user should be specified.
- A user should not be allowed to proceed without correcting an error.
- The system user should never get an operating system message or fatal error

## ER Diagram:

Entities are represented by rectangle.

Relation represented by diamond.

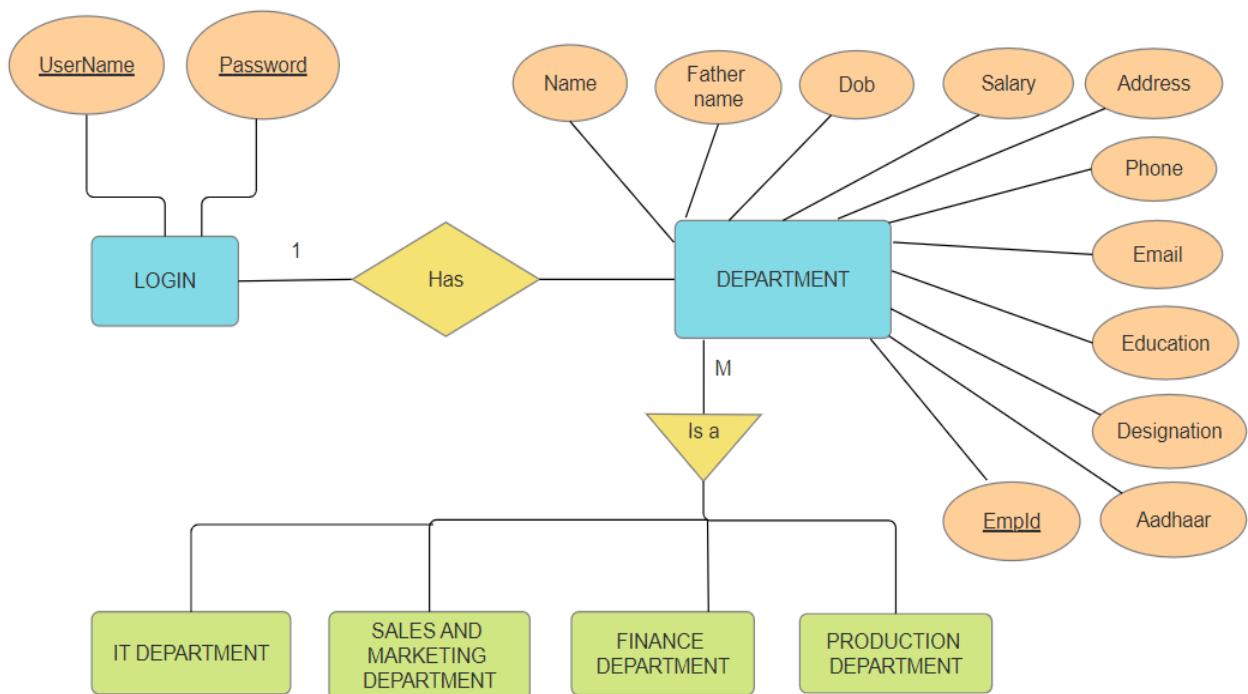
“Is a” relation represent the Specialization.

Ellipses symbolizes Attributes

It has following entities-

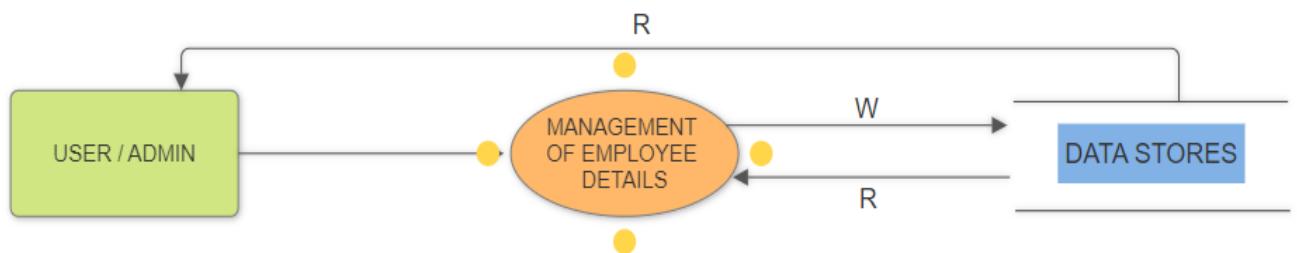
- Login
- Department
- IT Department
- Sales and Marketing Department
- Finance Department
- Production Department

**Note:-** “ we use the concept of Specialization to divide the Departments into 4 sub-division.”

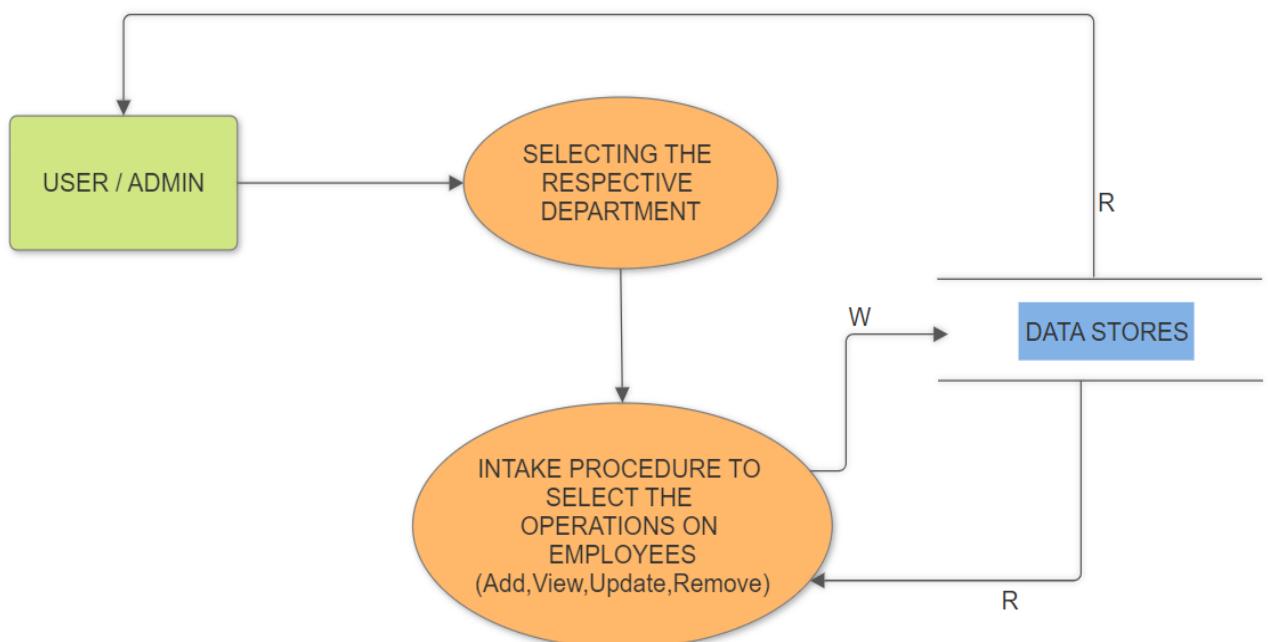


## DATA FLOW DIAGRAM

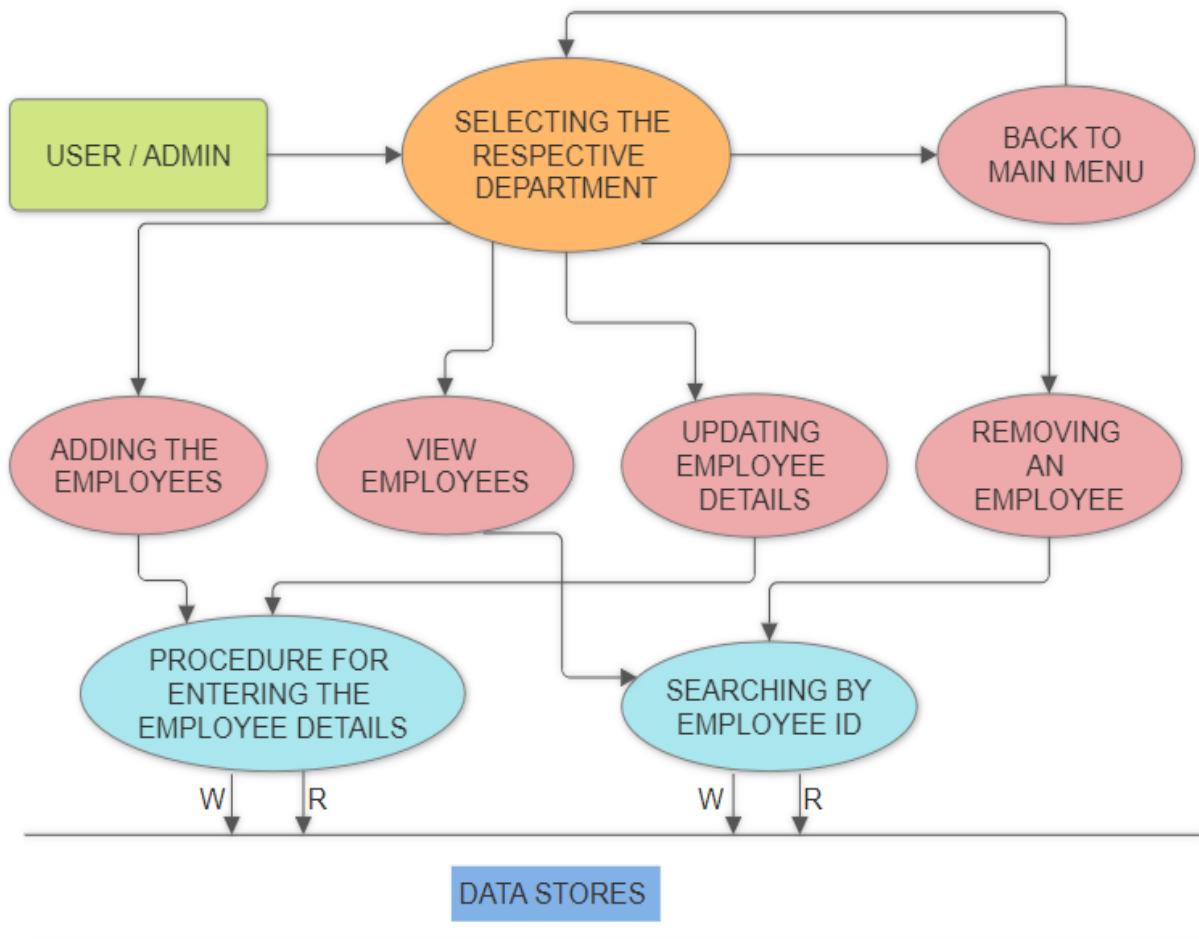
### DFD AT ZERO LEVEL



### DFD AT ONE LEVEL



## DFD AT TWO LEVEL



## I. Testing

Test ID	Test-Purpose	Test-Condition	Expected-Output	Output	Remark
TC1	Check Username & Password	If user details are not correct, display error message	Grant access to main dashboard.	Access granted to main dashboard	Test successful
TC2	To add new user to the system	If user already exists, error message should be displayed.	New user should be added.	New user added successfully	Test Successful
TC3	To view existing employee information	If employee exists, then information should be displayed, else error message should be displayed.	Employee information should be displayed.	Employee information displayed.	Test Successful
TC4	To remove an employee	If employee exists, then employee should be removed else error message should be displayed.	Employee should be removed.	Employee removed successfully.	Test Successful
TC5	Update employee information	If employee exists, then information should be updated.	Employee information should be updated.	Employee information updated successfully	Test Successful

## **Conclusion**

The goal of the initiative is to digitalize personnel databases in businesses and provide administrator's access to computers. Employees and administrators use software as an information system. The user can store his or her database safe and secure for an indefinite amount of time here. Adding, deleting, accessing, and changing employee information is simple and easy using the Employee Management System.

## **Future Scope of Employee Management System**

- It will help in storing and retrieval of employee information working in any organization.
- In a very short time the collection will be obvious, simple and sensible. It will reduce the cost of collecting the management and collection procedure.
- Our project based at Business Process automation
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To utilize resources in an efficient manner by increasing their productivity through automation
- The System generates information which can be used for various purposes
- It satisfies the user requirement
- Be easy to understand by the user and operator
- In future, there will be a better Graphical User Interface and there will be more features added to this system. If Graphical User Interface is improved then this system will be more user friendly and more features added will make this system a lot better and HR will be able to perform more operations.

## XVII. References

- 1) Renae Broderick, John W. Boudreau, "Human resource management, information technology, and the competitive edge", Academy of Management Executive, 1992 Vol. 6 No. 2
- 2) Julie Bulmash, "Human Resource Management and Technology", Chapter 3.
- 3) Ian Sommerville, "Software Engineering", 9<sup>th</sup> Edition, Addison-Wesley, 2011.
- 4) Juan Manuel Munoz Palacio, Information systems development methodologies for Data-driven Decision Support Systems, 2010.
- 5) Deitel, PJ & Deitel, HM, 2008, Internet & World Wide Web How to Program, Dorling Kindersley, India.
- 6) Google for problem solving.
- 7) <http://www.javapoint.com/java-tutorial>



# Chaudhary Charan Singh University, Meerut

Examination Session - 2023-2024

B.C.A. - (Semester 5)

Exam Category: Regular Main Exam

Exam Form #	230010462		Verification #	Submitted	
Exam Roll Number:	210613106184	Enrollment No. :	21017132	RegNo:	21G0110849
College Name:	I.I.M.T. COLLEGE OF MANAGEMENT				
Program Name	B.C.A.				
Caste/Category:	UNRESERVED	Sub-Category	NOT APPLICABLE		
Gender:	Male	Birthdate:	16-08-2002		
Nationality:	Indian	Religion:	Hindu		
N.A.D. ID. :					
Student Name:	PRIYAM SHARMA				
Father Name:	ALOK SHARMA				
Mother Name:	KARUNA SHARMA	Mobile:	9599833617		
Email:	ALOK196668@YAHOO.COM				
Perm. Address:	S/O: ALOK SHARMA, A-81, NEAR CHAURA MOD, SECTOR-22, NOIDA, GAUTAM BUDDHA				
Corr. Address:	S/O: ALOK SHARMA, A-81, NEAR CHAURA MOD, SECTOR-22, NOIDA, GAUTAM BUDDHA				
Courses / Class	B.C.A.	Exam / Board	Other	Year	Semester 4
Roll No	210613106184	M.Obt	421.0	Max	600.0



## Payment Details

Payment Status	Paid On	Payment Reference	Paid Amount
Success	21-11-2023	MCCSU7144194782	2357.0

## Fee Details

Fee Head	Amount
SPORTS FEE	70.00
EXAM FEE	800.00
Paper Code - 507	0.00
PROJECT FEE	1390.00
ROVER RANGER	10.00
ENROLLMENT FEE	0.00
MARKS FEE	80.00
CULTURAL ACTIVITY	7.00
DEGREE FEE	0.00



# Chaudhary Charan Singh University, Meerut

Examination Session - 2023-2024

B.C.A. - (Semester 5)

Exam Category: Regular Main Exam

Exam Form No :	230010462	Exam Roll No :	210613106184	Enrollment No :	21017132
----------------	-----------	----------------	--------------	-----------------	----------

Serial No	Paper id	Paper Code	Paper Name
1	SID_013748	501	Introduction to DBMS
2	SID_013749	502	Java Programming and Dynamic Webpage
3	SID_013750	503	Computer Network
4	SID_013751	504	Numerical Methods
5	SID_013752	505	Practical (DBMS)
6	SID_013753	506	Practical (Java Programming & Dynamic Webpage Design)
7	SID_013754	507	Minor Project
8	SID_013755	508	VivaVoice on Summer Training



## अभ्यर्थी की घोषणा

मैं

PRIYAM SHARMA

पुत्र/पुत्री

ALOK SHARMA

फॉर्म संख्या

230010462

शपथपूर्वक घोषणा करता/करती हूँ कि आवेदन पत्र में मेरे द्वारा अंकित समस्त विवरण पूर्णतया सत्य है तथा आवेदन पत्र के साथ संलग्न अंक पत्र, जाति प्रमाण पत्र इत्यादि की छाया प्रतियाँ वैध हैं। आवेदन पत्र में वर्णित सूचनाएं/प्रश्न पत्र अस्पष्ट अथवा अपूर्ण एवं गलत होने की दशा में अथवा अपेक्षित न्यूनतम योग्यता धारक न होने के कारण अहं न होने की दशा में विश्वविद्यालय द्वारा बिना कोई सुचना प्रेषित किये परीक्षा आवेदन पत्र/अभ्यर्थन निरस्त किये जाने पर मुझे कोई आपत्ति नहीं होगी। प्रवेश एवं परीक्षा सम्बन्धी नियमों में किसी भी समय किया गया प्रत्येक परिवर्तन मुझे सर्वमान्य होगा। परीक्षा में सम्मिलित हो जाने के उपरांत भी त्रुटि पाए जाने पर विश्वविद्यालय द्वारा मेरी सम्पूर्ण परीक्षा निरस्त किये जाने पर मुझे कोई आपत्ति नहीं होगी। मैं यह घोषणा करता हूँ/करती हूँ कि विगत परीक्षा वर्षों में अनुचित साधन प्रयोग करने के अपराध में दण्डित नहीं हुआ हूँ/हुई हूँ। परीक्षा एवं पाठ्यक्रम सम्बन्धी नियम तथा निर्देशों का मेरे द्वारा भलीं-भाँति अध्ययन कर लिया गया है। निर्धारित परीक्षा शुल्क (अन्यसमस्त शुल्क सहित) अन्य आवश्यक प्रपत्र माइग्रेशन आदि नामांकन हेतु एवं वर्तमान परीक्षा की अर्हता हेतु आवेदन पत्र के साथ संलग्न है। साथ ही मैं यह घोषणा करता हूँ/करती हूँ कि उपर्युक्त परीक्षा के अतिरिक्त मैं इस विश्वविद्यालय/किसी अन्य विश्वविद्यालय की किसी अन्य परीक्षा में सम्मिलित नहीं हो रहा हूँ/रही हूँ। मुझे विश्वविद्यालय द्वारा निर्धारित/परिवर्तित परीक्षा केंद्र पर परीक्षा देना स्वीकार्य है। मैं यह घोषणा करता हूँ/करती हूँ कि यदि परीक्षा सम्बन्धी कोई विवाद होगा तो उसका न्यायिक क्षेत्र मेरठ नगर स्थित न्यायालय या उच्च न्यायालय इलाहाबाद (प्रयागराज) होगा।

नोट-1 :— अभ्यर्थी के इस घोषणा पत्र पर हस्ताक्षर न होने पर वह परीक्षा में सम्मिलित होने का पात्र नहीं है।

नोट-2 :— परीक्षा फॉर्म की हार्ड-कॉपी निर्धारित समय सीमा के अन्दर सम्बंधित महाविद्यालय में जमा करना अनिवार्य है।

नोट-3 :— अभ्यर्थी द्वारा घोषित इस कोर्स का प्रवेश वर्ष — 2021

21 November, 2023

अभ्यर्थी का नाम (हिन्दी में)

दिनांक

अभ्यर्थी के हस्ताक्षर

[0613]-I.I.M.T. COLLEGE OF MANAGEMENT

## प्राचार्य/प्राचार्या का प्रमाण

प्रमाणित किया जाता है कि उपर्युक्त आवेदक/आवेदिका इस महाविद्यालय का/की संस्थागत/व्यक्तिगत छात्र/छात्रा है। मेरे ज्ञान तथा विश्वास में छात्र/छात्रा के विरुद्ध ऐसा कोई तथ्य नहीं है जो परीक्षा से वंचित करने योग्य हो एवं महाविद्यालय में प्रवेश विश्वविद्यालय की प्रवेश नियमावली के अनुसार लिया गया है। छात्र/छात्रा द्वारा प्रस्तुत विवरण विश्वसनीय है। यह भी प्रमाणित किया जाता है कि अभिलेखानुसार, वर्तमान उपस्थिति के आधार पर यह अपेक्षा की जा सकती है कि संस्थागत अभ्यर्थी अपनी वांछित उपस्थिति नियमों के अनुरूप सत्र कि शेष अवधि में प्राप्त करने में सक्षम है तथापि परीक्षा प्रारंभ होने से पूर्व उपस्थित पूर्ण न होने की सुचना विश्वविद्यालय को पृथक से प्रेषित की जाएगी। यह भी प्रमाणित किया जाता है कि छात्र/छात्रा सम्बंधित परीक्षा में बैठने के लिए अहं है।

दिनांक

हस्ताक्षर आवेदन पत्र जांचकर्ता  
(महाविद्यालय स्तर पर)

प्राचार्य/प्राचार्या/वि.वि विभागाध्यक्ष  
के हस्ताक्षर (मोहर सहित)