### Dr.A.P.J Abdul Kalam Technical University Lucknow Submitted In Partial fulfillment of the requirement for the award of the

### Degree In

### 

### MASTER OF COMPUTER APPLICATION (MCA)

### (2021-2023) 3rd Semester

# LAL BAHADUR SHASTRI INSTITUTE OF MANAGEMENT AND DEVLOPMENT STUDIES LUCKNOW - 226026



**Mini Project**

On

**“IT Office Employees Record System”**

## **Submitted By** **Under the Guidance of**

## **Name: SUNEET YADAV Ms**. **JAISHREE SRIVASTAVA(HOD)**

**Roll No :- 2100590140049**

# ACKNOWLEDGEMENT

I **SUNEET YADAV** student of MCA hereby declare that the Project titled **“IT Office Employees Record System”** which is submitted by me to **Ms**. **Jaishree Srivastava Ma’am(HOD),Dr. Ina Kapoor(Deputy HOD) and Mr.Pankaj Kumar(Assistant Professor) Dept.Computer Science & Engineering of College** LAL BAHADUR SHASTRI INSTITUTE OF MANAGEMENT AND DEVLOPENT STUDIES, LUCKNOW,India and Thanking them for their generous Guidance ,help and usefull suggestions.

It is high privilege for me to express my deep sense of gratitude to those entire faculty Members who helped me in the completion of the project, specially my internal guide **Ms**. **Jaishree Srivatava Ma’am(HOD),**who was always there at hour of need. My special thanks to all other faculty members, Batch mate & Seniors of my College for helping me in the completion of project work and its report submission.

## **Place-** LUCKNOW **Student Name-** SUNEET YADAV

**Date-** 27-02-2023 **Enrollment No**- 2100590140049



# GUIDE CERTFICATE

I hereby certify that the Seminar Report by student of **MCA 3RD SEMESTER ROLLNO 2100590140049** with title **“IT Office Employees Record System ”** which is submitted to **LAL BAHADUR SHASTRI INSTITUTE OF MANAGEMENT AND DEVLOPMENT STUDIES, LUCKNOW,** in partial fulfillment of requirement for the award of the degree of MCA is an original contribution with existing knowledge and faithful record of work carried out by him/her under my guidance and supervision and to the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

**Place** Lucknow MR. Pankaj Kumar

**Date:-** 20-02-2023: Assistant Professor

**Certificate From Internshala**



**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **S. No.** | **Topic** |
| 1. | Abstract |
| 2. | Chapter-1: Introduction |
| 3. | Chapter-2: Objective |
| 4. | Chapter-3: Hardware and Software Require |
| 5. | Chapter-4: Scope of the project |
| 6. | Chapter-5:Data Flow Diagram and E-R Diagram |
| 7. | Chapter-6: Methodology |
| 8. | Chapter-7: Feasibility Study |
| 9. | Chapter-8: Implementation and Results |
| 10. | Chapter-8: Testing |
| 11. | Chapter-9: Conclusion |
| 12. | Chapter-10: Future Scope |
| 13. | Chapter -11: Bibliography |

# 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.*

# Introduction

Everything has been digitised 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 "IT OFFICE EMPLOYEE RECORD SYSTEM" was created to address the issues that plagued the previous manual system. This programme 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.

# 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.

**HARDWARE AND SOFTWARE REQUIREMENTS**

**Software Required**

* For Creating IT Office Employees Record System IDE is required for coding and run GUI based application.
* I have use Apache Net Beans IDE 13 and for Database use MYSQL.

**Hardware Required**

A Personal Computer A system with at least Pentium dual core processor, 4GB RAM and 128 GB storage.

* + - Operating System : Window, Ubuntu etc.
    - Databases : MySql

# Scope of Project

Employees are the backbone of any company, management of employee performance plays a major role in deciding the success of the organization. The service center has a problem in management of employee performance.

The current system is paper and individual based keeping notes on irregular basis and keeping old fashioned matter which is not accessible for quick analysis.

To understand better following are the areas which will serve the solution scope.

**Problem Domain**: The problem domain is the computerization of an Employee Management System that accessed can be online by the administrator and the Stockholder Such as the manager.

**The current System:** This System is used to store the information of employee of organization.

Current system works on trust and data entry is being used on paper is manual and can be altered, employees use behavioral and personal realtiaons to take attendance benefits, thusresulting the center in final output shrinkage. Register and documents often misplaced or lost or even can get fire.

**DATA SETS**

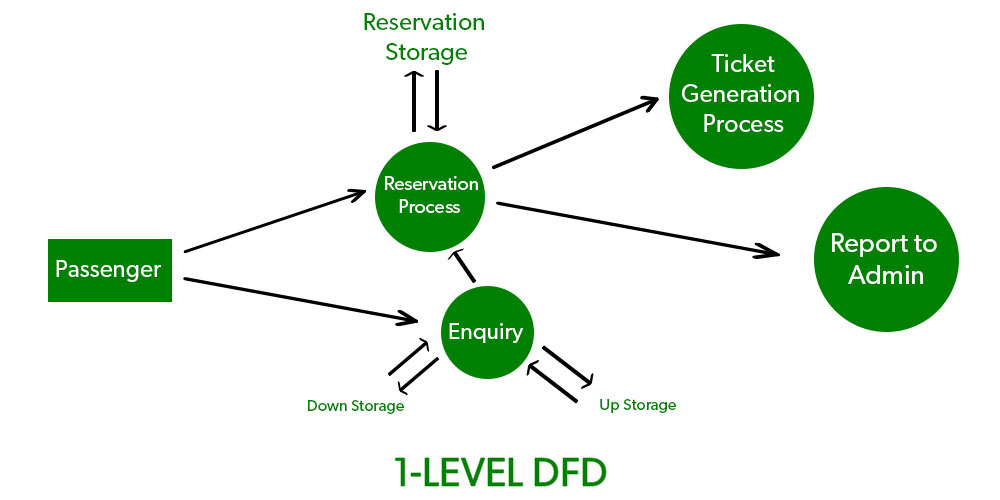
**Dashboard:**  View the populated data based on criteria or filters such as total employees count, employees present or absent based on shift, expected salary amount to be disbursed.

**Login: For** Login page to admin.

**Homepage:** Thepage where all the Buttons are showed.

**Data Flow Diagram**

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. It is a graphical tool, useful for communicating with users, managers and other personnel. it is useful for analyzing existing as well as proposed system. It provides an overview of What data is system processes.



• What transformation is performed?

• What data are stored?

• What results are produced, etc?

Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

* **Process** Input to output transformation in a system takes place because of process function. The symbols of a process are rectangular with rounded corners, oval, rectangle or a circle. The process is named a short sentence, in one word or a phrase to express its essence
* **Data Flow** Data flow describes the information transferring between different parts of the systems. The arrow symbol is the symbol of data flow. A relatable name should be given to the flow to determine the information which is being moved. Data flow also represents material along with information that is being moved. Material shifts are modeled in systems that are not merely informative. A given flow should only transfer a single type of information. The direction of flow is represented by the arrow which can also be bi-directional.
* **Warehouse** The data is stored in the warehouse for later use. Two horizontal lines represent the symbol of the store. The warehouse is simply not restricted to being a data file rather it can be anything like a folder with documents, an optical disc, a filing cabinet. The data warehouse can be viewed independent of its implementation. When the data flow from the warehouse it is considered as data reading and when data flows to the warehouse it is called data entry or data updating.
* **Terminator** The Terminator is an external entity that stands outside of the system and communicates with the system. It can be, for example, organizations like banks, groups of people like customers or different departments of the same organization, which is not a part of the model system and is an external entity. Modeled systems also communicate with terminator.

***Rules for creating DFD***

* The name of the entity should be easy and understandable without any extra assistance(like comments).
* The processes should be numbered or put in ordered list to be referred easily.
* The DFD should maintain consistency across all the DFD levels.
* A single DFD can have maximum processes up to 9 and minimum 3 processes.

Symbols Used in DFD

* **Square Box:** A square box defines source or destination of the system. It is also called entity. It is represented by rectangle.

**Arrow or Line:** An arrow identifies the data flow i.e. it gives information to the data that is in motion

* **Circle or bubble chart:** It represents as a process that gives us information. It is also called processing box

.

* **Open Rectangle:** An open rectangle is a data store. In this data is store either temporary or permanently.

Levels of DFD

DFD uses hierarchy to maintain transparency thus multilevel DFD’s can be created. Levels of DFD are as follows:

* 0-level DFD
* 1-level DFD:
* 2-level DFD:

Advantages of DFD

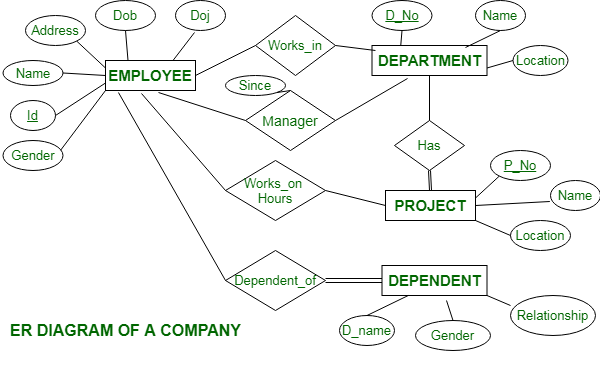
* It helps us to understand the functioning and the limits of a system.
* It is a graphical representation which is very easy to understand as it helps visualize contents.
* Data Flow Diagram represent detailed and well explained diagram of system components.
* It is used as the part of system documentation file.
* Data Flow Diagrams can be understood by both technical and nontechnicalperson because they are very easy to understand.

Disadvantages of DFD

* At times DFD can confuse the programmers regarding the system.
* Data Flow Diagram takes long time to be generated, and many times due to this reasons analysts are denied permission to work on it.

**ENTITY RELATIONSHIP(ER) DIGRAM**

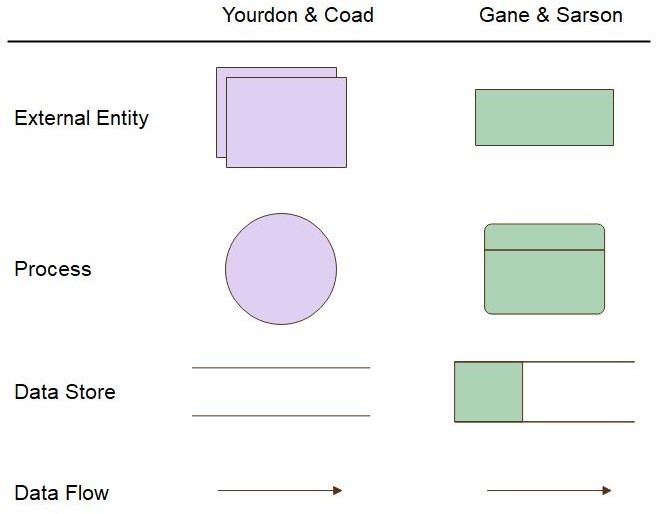
An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.



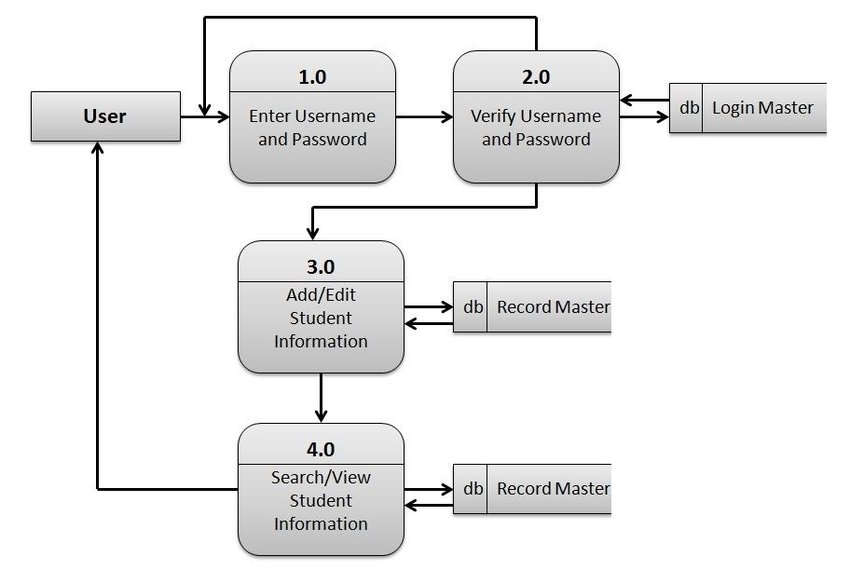
## **Data Flow Diagram Symbols**

DFD symbols are consistent notations that depict a system or a process. It entails the use of short-text labels, arrows, circles and rectangles to describe data flow direction. Also forming part of DFDs are varied sub-processes, data storage points, and data inputs and outputs.

A data flow diagram has four basic elements. The elements include external entities, data stores, processes, and data flows. The elements are best represented by the two main methods of notation used in DFDs – **Yourdon & Coad**, and **Gane & Sarson**. DFD symbols vary slightly depending on methodology. Even so, the basic ideas remain the same.



**Data Flow Diagram of IT Office Employee Record Project**



Search/View Employee Information

Enter Username and Password to Login

Add/Edit /Remove

Employee Informatio

Verify Username and Password

# 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.

**APACHE NET BEANS-** Net Beans is a Java-based integrated development environment (IDE). Net Beans enables the creation of applications using a set of modular software components known as modules. Net Beans 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 Net Beans-based applications, including the Net Beans 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.

# 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.

1. **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?

1. **Economic Feasibility**

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

# 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. **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.

**Code For Module Login**

**public class Login extends JFrame implements ActionListener{**

**JTextField tfusername;**

**JPasswordField tfpassword;**

**Checkbox ckbox;**

**Login() {**

**getContentPane().setBackground(Color.WHITE);**

**setLayout(null);**

**JLabel lblusername=new JLabel("USERNAME");**

**lblusername.setBounds(40,20,100,30);**

**lblusername.setForeground(Color.BLACK);**

**add(lblusername);**

**tfusername= new JTextField();**

**tfusername.setBounds(150,20,100,30);**

**add(tfusername);**

**JLabel lblpassword=new JLabel("PASSWORD");**

**lblpassword.setForeground(Color.BLACK);**

**lblpassword.setBounds(40,70,100,30);**

**add(lblpassword);**

**tfpassword= new JPasswordField();**

**tfpassword.setBounds(150,70,100,30);**

**add(tfpassword);**

**JButton login=new JButton ("LOGIN");**

**login.setBounds(80,140,140,70);**

**login.setBackground(Color.BLACK);**

**login.setForeground(Color.WHITE);**

**login.addActionListener(this);**

**add(login);**

**ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("image/login.jpg"));**

**Image i2=i1.getImage().getScaledInstance(650,600,Image.SCALE\_DEFAULT);**

**ImageIcon i3=new ImageIcon(i2);**

**JLabel image=new JLabel(i3);**

**//cahnge image**

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

**add(image);**

**setSize(600,450);**

**setLocation(450,200);**

**setVisible(true);**

**}**

**public void actionPerformed(ActionEvent ae)**

**{ try{**

**String username=tfusername.getText();**

**String password=tfpassword.getText();**

**cone c=new cone();**

**String query = "select \* from login where username= '"+username+"' and password = '"+password+"'";**

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

**if(rs.next()){**

**setVisible(false);**

**new Home();**

**}else**

**{JOptionPane.showMessageDialog(null,"Invalid user name and passsword");} } }**

**catch(Exception e){**

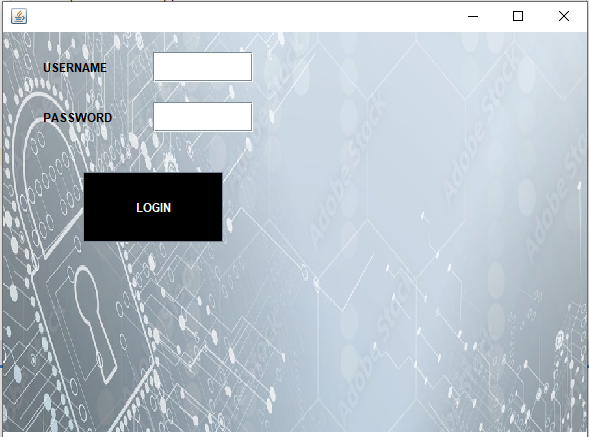
**e.printStackTrace();**

**}}**

**public static void main (String args[])**

**{**

**new Login(); }}**

****

**Fig. 1**

1. **Main Dashboard –** After login in, user is directed to the main dashboard of this system where user can perform various operations like adding an employee, deleting an employee.

**Code For Main Dashboard**

**package employee.managment.system;**

**import java.awt.\*;**

**import javax.swing.\*;**

**import java.awt.event.\*;**

**public class Home extends JFrame implements ActionListener {**

**JButton add,view,update,remove,exit;**

**Home(){**

**setLayout(null);**

**ImageIcon**

**i1=new ImageIcon(ClassLoader.getSystemResource("image/view.jpg"));**

**i2=i1.getImage().getScaledInstance(1110,700,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("IT Office Employees Record System");**

**heading.setBounds(550,20,500,40);**

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

**image.add(heading);**

**add= new JButton("Add Employee");**

**add.setBounds(650,80,150,40);**

**add.addActionListener(this);**

**image.add(add);**

**view= new JButton("View Employees");**

**view.setBounds(820,80,150,40);**

**view.addActionListener(this);**

**image.add(view);**

**update= new JButton("Update Employee");**

**update.setBounds(650,140,150,40);**

**update.addActionListener(this);**

**image.add(update);**

**remove= new JButton("Remove Employee");**

**remove.setBounds(820,140,150,40);**

**remove.addActionListener(this);**

**image.add(remove);**

**exit= new JButton("Exit");**

**exit.setBounds(750,200,150,40);**

**exit.addActionListener(this);**

**image.add(exit);**

**setSize(1120,630);**

**setLocation(150,100);**

**setVisible(true);**

**}**

**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)**

**{**

**new ViewEmployee();**

**}**

**else if(ae.getSource()==exit)**

**{**

**setVisible(false);**

**}**

**else {**

**new RemoveEmployee();}**

**}**

**public static void main (String args[])**

**{ new Home();**

**}}**



**Fig. 2**

1. **Add employee** – Here user have to enter all the required credentials to add a new employee to the system.

**Code For AddEmployee Module**

**package employee.managment.system;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.sql.\*;**

**import com.toedter.calendar.JDateChooser;**

**import java.util.\*;**

**public class AddEmployee extends JFrame implements ActionListener{**

**Random ran=new Random();**

**int number =ran.nextInt(999999);**

**JTextFieldtfname,tffname,tfemail,tfaddress,tfaadhar,tfphone,tfdesignation,tfsalary;**

**JComboBox cbeducation;**

**JButton add,back;**

**JDateChooser dcdob;**

**JLabel lblempid,lblempID;**

**AddEmployee(){**

**getContentPane().setBackground(Color.WHITE);**

**setLayout(null);**

**JLabel heading =new JLabel("Add Employee Detail");**

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

**heading.setFont(new Font("SAN\_SERIF",Font.BOLD,25));**

**add(heading);**

**JLabel name=new JLabel("Name");**

**name.setBounds(90,150,150,30);**

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

**add(name);**

**tfname=new JTextField();**

**tfname.setBounds(200,150,150,30);**

**add(tfname);**

**JLabel fname=new JLabel("Father name");**

**fname.setBounds(400,150,150,30);**

**fname.setFont(new Font("serif",Font.BOLD,20));**

**add(fname);**

**tffname=new JTextField();**

**tffname.setBounds(550,150,150,30);**

**add(tffname);**

**JLabel lbldob=new JLabel("Date of Birth");**

**lbldob.setBounds(90,200,150,30);**

**lbldob.setFont(new Font("serif",Font.BOLD,20));**

**add(lbldob);**

**dcdob =new JDateChooser();**

**dcdob.setBounds(200,200,150,30);**

**add(dcdob);**

**JLabel labelsalary=new JLabel("Salary");**

**labelsalary.setBounds(400,200,150,30);**

**labelsalary.setFont(new Font("serif",Font.BOLD,20));**

**add(labelsalary);**

**tfsalary=new JTextField();**

**tfsalary.setBounds(550,200,150,30);**

**add(tfsalary);**

**JLabel lbladdress=new JLabel("Address");**

**lbladdress.setBounds(90,250,150,30);**

**lbladdress.setFont(new Font("serif",Font.BOLD,20));**

**add(lbladdress);**

**tfaddress=new JTextField();**

**tfaddress.setBounds(200,250,150,30);**

**add(tfaddress);**

**JLabel lblphone=new JLabel("Mobile Number");**

**lblphone.setBounds(400,250,150,30);**

**lblphone.setFont(new Font("serif",Font.BOLD,20));**

**add(lblphone);**

**tfphone=new JTextField();**

**tfphone.setBounds(550,250,150,30);**

**add(tfphone);**

**JLabel lblemail=new JLabel("Email");**

**lblemail.setBounds(90,300,150,30);**

**lblemail.setFont(new Font("serif",Font.BOLD,20));**

**add(lblemail);**

**tfemail=new JTextField();**

**tfemail.setBounds(200,300,150,30);**

**add(tfemail);**

**JLabel lbleducation=new JLabel("Education");**

**lbleducation.setBounds(400,300,150,30);**

**lbleducation.setFont(new Font("serif",Font.BOLD,20));**

**add(lbleducation);**

**Stringcourses[]={"SELECT","BBA","BCA","MBA","MCA","BSc"," BSc(CS)","MA","BA","B.TECH","M.TECH","MSc","other"};**

**cbeducation=new JComboBox(courses);**

**cbeducation.setBackground(Color.WHITE);**

**cbeducation.setBounds(550,300,150,30);**

**add(cbeducation);**

**JLabel lbldesignation=new JLabel("Designation");**

**lbldesignation.setBounds(90,350,150,30);**

**lbldesignation.setFont(new Font("serif",Font.BOLD,20));**

**add(lbldesignation);**

**tfdesignation=new JTextField();**

**tfdesignation.setBounds(200,350,150,30);**

**add(tfdesignation);**

**JLabel lblaadhar=new JLabel("Aadhar");**

**lblaadhar.setBounds(400,350,150,30);**

**lblaadhar.setFont(new Font("serif",Font.BOLD,20));**

**add(lblaadhar);**

**tfaadhar=new JTextField();**

**tfaadhar.setBounds(550,350,150,30);**

**add(tfaadhar);**

**lblempid=new JLabel("EmployeeID");**

**lblempid.setBounds(90,400,150,30);**

**lblempid.setFont(new Font("serif",Font.BOLD,20));**

**add(lblempid);**

**lblempID=new JLabel("" + number);**

**lblempID.setBounds(200,400,150,30);**

**lblempID.setFont(new Font("serif",Font.BOLD,20));**

**add(lblempID);**

**add= new JButton("ADD");**

**add.setBounds(250,550,150,40);**

**add.setBackground(Color.gray);**

**add.setForeground(Color.BLUE);**

**add.addActionListener(this);**

**add(add);**

**back= new JButton("BACK");**

**back.setBounds(450,550,150,40);**

**back.setBackground(Color.gray);**

**back.setForeground(Color.BLUE);**

**back.addActionListener(this);**

**add(back);**

**ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("image/addemp.jpg"));**

**Image i2=i1.getImage().getScaledInstance(1110,700,Image.SCALE\_DEFAULT);**

**ImageIcon i3=new ImageIcon(i2);**

**JLabel image=new JLabel(i3);**

**image.setBounds(0,0,1120,630);**

**add(image);**

**setSize(900,700);**

**setLocation(300,50);**

**setVisible(true);**

**}**

**public void actionPerformed(ActionEvent ae){**

**if (ae.getSource()==add )**

**{**

**String name= tfname.getText();**

**Stringfname=tffname.getText();**

**Stringdob=((JTextField)dcdob.getDateEditor().getUiComponent()).getText();**

**String salary=tfsalary.getText();**

**String address=tfaddress.getText();**

**String phone=tfphone.getText();**

**String email=tfemail.getText();**

**String education=(String)cbeducation.getSelectedItem();**

**String designation=tfdesignation.getText();**

**String aadhar=tfaadhar.getText();**

**String empId=lblempID.getText() ;**

**try{**

**cone conn =new cone();**

**String query="insert into employee values('"+name+"','"+fname+"','"+dob+"','"+salary+"','"+address+"','"+phone+"','"+email+"','"+education+"','"+designation+"','"+aadhar+"','"+empId+"')";**

**conn.s.executeUpdate(query);**

**JOptionPane.showMessageDialog(null,"Details added successfully");**

**setVisible(false);**

**new Home();**

**}catch(Exception e){**

**e.printStackTrace();**

**}**

**}else**

**{**

**setVisible(false);**

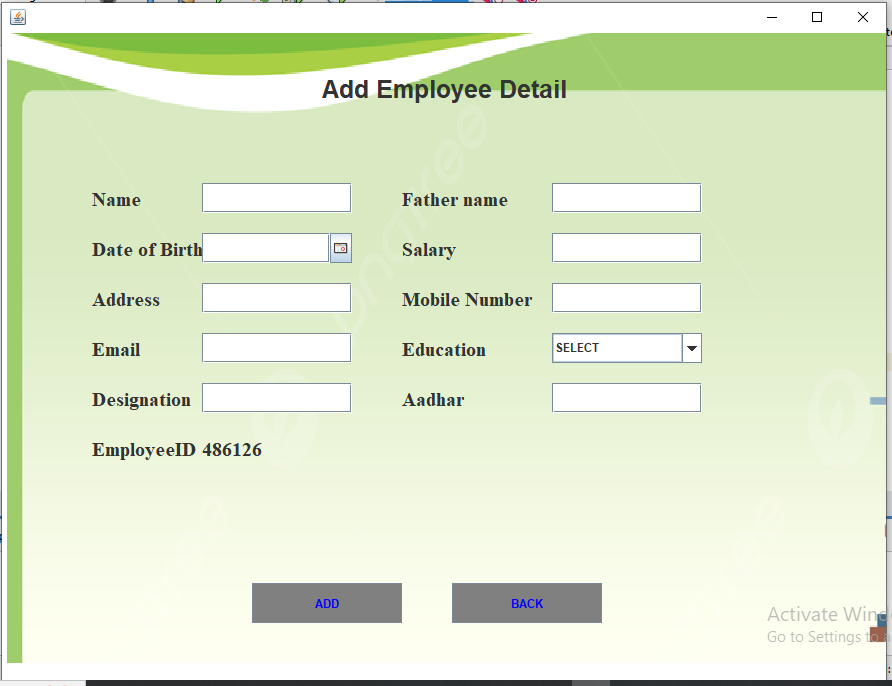
**new Home();}**

**} public static void main (String args[])**

**{**

**new AddEmployee();**

**} }**



**Fig. 3**

1. **Remove employee** – User has to enter the employee id in order to delete his information from the system.

**Code For RemoveEmployee Module**

**package employee.managment.system;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.sql.\*;**

**public class RemoveEmployee extends JFrame implements ActionListener{**

**Choice cempid;**

**JButton Delete,Back;**

**RemoveEmployee(){**

**setLayout(null);**

**getContentPane().setBackground(Color.WHITE);**

**JLabel labelempid=new JLabel("Emoloyee ID");**

**labelempid.setBounds(50,50,100,30);**

**add(labelempid);**

**cempid=new Choice();**

**cempid.setBounds(200,50,150,30);**

**add(cempid);**

**try{**

**cone c =new cone();**

**String query="select \* from employee";**

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

**while(rs.next())**

**{**

**cempid.add(rs.getString("empid"));**

**}**

**}catch(Exception e){**

**e.printStackTrace();**

**}**

**JLabel labelname=new JLabel("Name");**

**labelname.setBounds(50,100,100,30);**

**add(labelname);**

**JLabel lblname=new JLabel();**

**lblname.setBounds(200,100,100,30);**

**add(lblname);**

**JLabel lbldob=new JLabel("DoB");**

**lbldob.setBounds(50,150,100,30);**

**add(lbldob);**

**JLabel labeldob=new JLabel();**

**labeldob.setBounds(200,150,100,30);**

**add(labeldob);**

**JLabel lblphone=new JLabel("Phone");**

**lblphone.setBounds(50,200,100,30);**

**add(lblphone);**

**JLabel labelphone=new JLabel();**

**labelphone.setBounds(200,200,100,30);**

**add(labelphone);**

**JLabel labelfname=new JLabel("FName");**

**labelfname.setBounds(50,250,100,30);**

**add(labelfname);**

**JLabel lblfname=new JLabel();**

**lblfname.setBounds(200,250,100,30);**

**add(lblfname);**

**try{**

**cone c =new cone();**

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

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

**while(rs.next())**

**{lblname.setText(rs.getString("name"));**

**labeldob.setText(rs.getString("dob"));**

**labelphone.setText(rs.getString("phone"));**

**lblfname.setText(rs.getString("fname"));**

**}}catch(Exception e){**

**e.printStackTrace();**

**}**

**cempid.addItemListener(new ItemListener() {**

**public void itemStateChanged(ItemEvent ie){**

**try{**

**cone c =new cone();**

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

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

**while(rs.next())**

**{lblname.setText(rs.getString("name"));**

**labeldob.setText(rs.getString("dob"));**

**labelphone.setText(rs.getString("phone"));**

**lblfname.setText(rs.getString("fname"));**

**}**

**}catch(Exception e){**

**e.printStackTrace();**

**}}**

**});**

**Delete =new JButton("Delete");**

**Delete.setBounds(80,300,100,30);**

**Delete.setForeground(Color.WHITE);**

**Delete.setBackground(Color.BLACK);**

**Delete.addActionListener(this);**

**add(Delete);**

**Back =new JButton("Back");**

**Back.setBounds(220,300,100,30);**

**Back.setForeground(Color.WHITE);**

**Back.setBackground(Color.BLACK);**

**Back.addActionListener(this);**

**add(Back);**

**ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("image/deelt.jpg"));**

**Image i2=i1.getImage().getScaledInstance(600,500,Image.SCALE\_DEFAULT);**

**ImageIcon i3=new ImageIcon(i2);**

**JLabel image=new JLabel(i3);**

**image.setBounds(0,0,600,500);**

**add(image);**

**setSize(600,500);**

**setLocation(350,150);**

**setVisible(true);**

**}**

**public void actionPerformed(ActionEvent ae)**

**{**

**if(ae.getSource()==Delete){**

**try{**

**cone c=new cone();**

**String query="delete from employee where empId='"+cempid.getSelectedItem()+"'";**

**c.s.executeUpdate(query);**

**JOptionPane.showMessageDialog(null,"Employee Information delted succesfully");**

**setVisible(false);**

**new Home();**

**}**

**catch(Exception e){**

**e.printStackTrace();**

**}}**

**else{**

**setVisible(false);**

**new Home();**

**}**

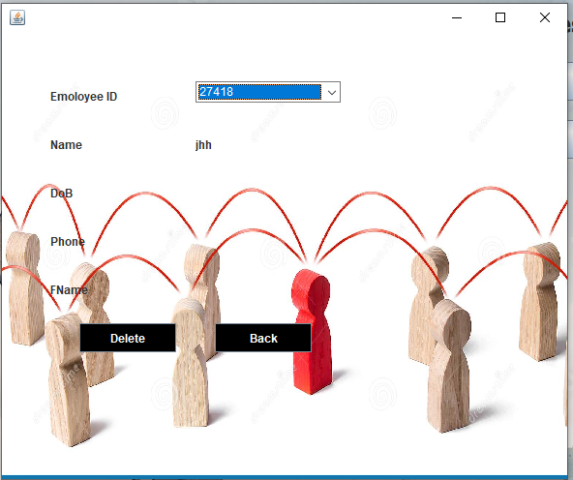
**}**

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

**new RemoveEmployee();**

**}**

}



**Fig.4**

1. **View and update employee** – In order to view and update employee information, the user have to enter employee ID.

**Code For View and Update Employee Code**

**package employee.managment.system;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.sql.\*;**

**import java.util.\*;**

**import net.proteanit.sql.DbUtils;**

**public class ViewEmployee extends JFrame implements ActionListener{**

**JTable table;**

**Choice cemployeeid;**

**JButton search,print,update,back;**

**ViewEmployee(){**

**getContentPane().setBackground(Color.CYAN);**

**setLayout(null);**

**JLabel searchlbl=new JLabel("Search by Employee Id");**

**searchlbl.setBounds(20,20,150,20);**

**add(searchlbl);**

**cemployeeid = new Choice();**

**cemployeeid.setBounds(200,20,150,20);**

**add(cemployeeid);**

**try{**

**cone c=new cone();**

**ResultSet rs= c.s.executeQuery("select \* from employee");**

**while(rs.next()){**

**cemployeeid.add(rs.getString("empid"));**

**}**

**}catch(Exception e)**

**{**

**e.printStackTrace();**

**}**

**table=new JTable();**

**try{**

**cone c=new cone();**

**ResultSet rs= c.s.executeQuery("select \* from employee");**

**table.setModel(DbUtils.resultSetToTableModel(rs));**

**while(rs.next()){**

**cemployeeid.add(rs.getString("empid"));**

**}**

**}catch(Exception e)**

**{**

**e.printStackTrace();**

**}**

**JScrollPane jsp=new JScrollPane(table);**

**jsp.setBounds(10,100,900,600);**

**add(jsp);**

**search=new JButton("Search");**

**search.setBounds(20,70,80,20);**

**search.addActionListener(this);**

**add(search);**

**print=new JButton("Print");**

**print.setBounds(120,70,80,20);**

**print.addActionListener(this);**

**add(print);**

**update=new JButton("Update");**

**update.setBounds(220,70,80,20);**

**update.addActionListener(this);**

**add(update);**

**back=new JButton("Back");**

**back.setBounds(320,70,80,20);**

**back.addActionListener(this);**

**add(back);**

**setSize(930,700);**

**setLocation(200,100);**

**setVisible(true);**

**}**

**public void actionPerformed(ActionEvent ae){**

**if(ae.getSource()==search)**

**{**

**String query =("select \* from employee where empId='"+cemployeeid.getSelectedItem()+"'");**

**try{**

**cone c=new cone();**

**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(Exception e){**

**e.printStackTrace();**

**}**

**}**

**else if(ae.getSource()==update)**

**{**

**setVisible(false);**

**new UpdateEmployee(cemployeeid.getSelectedItem());**

**}else**

**{**

**setVisible(false);**

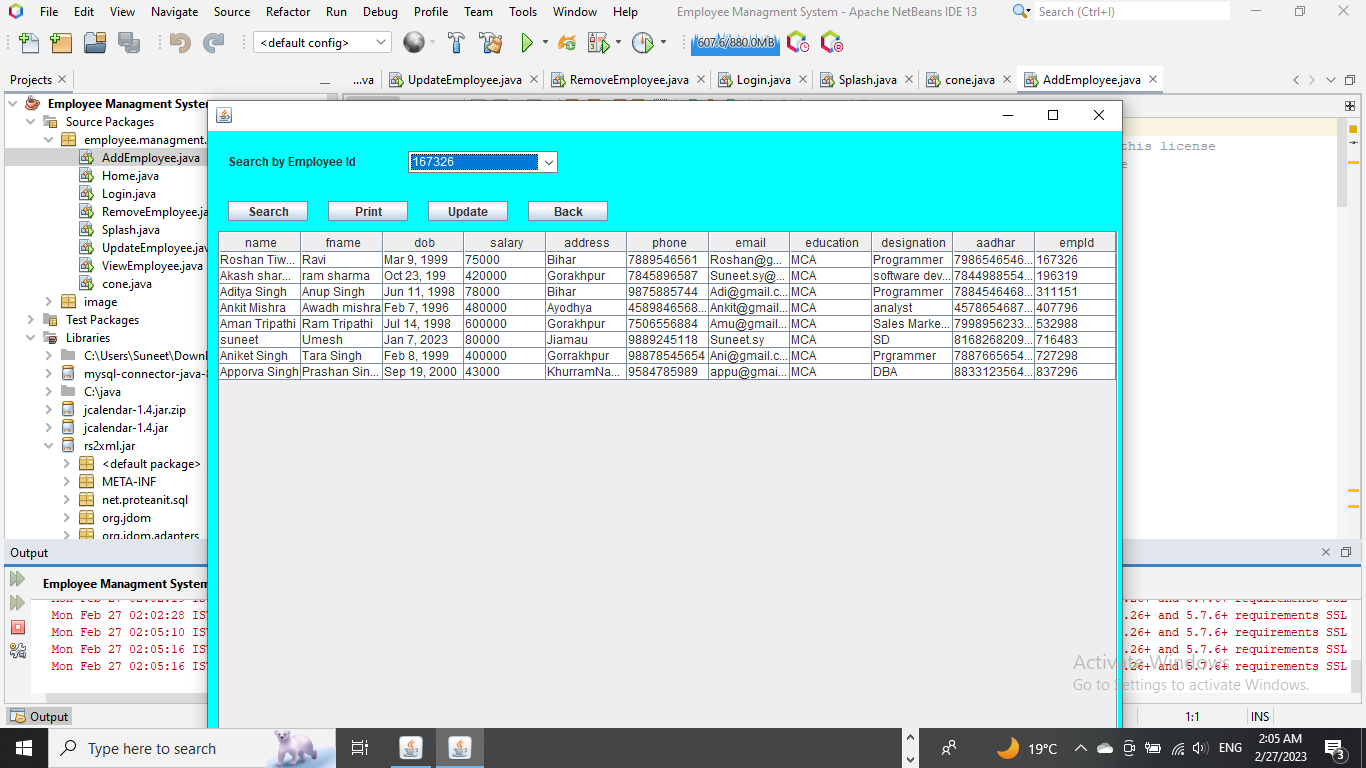
**new Home();**

**}}**

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

new ViewEmployee();

}}



**Fig. 5**

**Code For UpdateEmployee Module**

**public class UpdateEmployee extends JFrame implements ActionListener{**

**JTextField tffname,tfeducation,tfemail,tfaddress,tfphone,tfdesignation,tfsalary;**

**JButton add,back;**

**String empId;**

**JLabel lblempid,lblempID;**

**UpdateEmployee(String empId){**

**this.empId=empId;**

**getContentPane().setBackground(Color.WHITE);**

**setLayout(null);**

**JLabel heading =new JLabel("Update Employee Details");**

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

**heading.setFont(new Font("SAN\_SERIF",Font.BOLD,25));**

**add(heading);**

**JLabel name=new JLabel("Name");**

**name.setBounds(90,150,150,30);**

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

**add(name);**

**JLabel lblname=new JLabel();**

**lblname.setBounds(200,150,150,30);**

**add(lblname);**

**JLabel fname=new JLabel("Father name");**

**fname.setBounds(400,150,150,30);**

**fname.setFont(new Font("serif",Font.BOLD,20));**

**add(fname);**

**tffname=new JTextField();**

**tffname.setBounds(550,150,150,30);**

**add(tffname);**

**JLabel lbldob=new JLabel("Date of Birth");**

**lbldob.setBounds(90,200,150,30);**

**lbldob.setFont(new Font("serif",Font.BOLD,20));**

**add(lbldob);**

**JLabel lbeldob = new JLabel();**

**lbeldob.setBounds(200,200,150,30);**

**add(lbeldob);**

**JLabel labelsalary=new JLabel("Salary");**

**labelsalary.setBounds(400,200,150,30);**

**labelsalary.setFont(new Font("serif",Font.BOLD,20));**

**add(labelsalary);**

**tfsalary=new JTextField();**

**tfsalary.setBounds(550,200,150,30);**

**add(tfsalary);**

**JLabel lbladdress=new JLabel("Address");**

**lbladdress.setBounds(90,250,150,30);**

**lbladdress.setFont(new Font("serif",Font.BOLD,20));**

**add(lbladdress);**

**tfaddress=new JTextField();**

**tfaddress.setBounds(200,250,150,30);**

**add(tfaddress);**

**JLabel lblphone=new JLabel("Mobile Number");**

**lblphone.setBounds(400,250,150,30);**

**lblphone.setFont(new Font("serif",Font.BOLD,20));**

**add(lblphone);**

**tfphone=new JTextField();**

**tfphone.setBounds(550,250,150,30);**

**add(tfphone);**

**JLabel lblemail=new JLabel("Email");**

**lblemail.setBounds(90,300,150,30);**

**lblemail.setFont(new Font("serif",Font.BOLD,20));**

**add(lblemail);**

**tfemail=new JTextField();**

**tfemail.setBounds(200,300,150,30);**

**add(tfemail);**

**JLabel lbleducation=new JLabel("Education");**

**lbleducation.setBounds(400,300,150,30);**

**lbleducation.setFont(new Font("serif",Font.BOLD,20));**

**add(lbleducation);**

**tfeducation = new JTextField();**

**tfeducation.setBackground(Color.WHITE);**

**tfeducation.setBounds(550,300,150,30);**

**add(tfeducation);**

**JLabel lbldesignation=new JLabel("Designation");**

**lbldesignation.setBounds(90,350,150,30);**

**lbldesignation.setFont(new Font("serif",Font.BOLD,20));**

**add(lbldesignation);**

**tfdesignation=new JTextField();**

**tfdesignation.setBounds(200,350,150,30);**

**add(tfdesignation);**

**JLabel lblaadhar=new JLabel("Aadhar");**

**lblaadhar.setBounds(400,350,150,30);**

**lblaadhar.setFont(new Font("serif",Font.BOLD,20));**

**add(lblaadhar);**

**JLabel lbelaadhar=new JLabel();**

**lbelaadhar.setBounds(550,350,150,30);**

**add(lbelaadhar);**

**lblempid=new JLabel("EmployeeID");**

**lblempid.setBounds(90,400,150,30);**

**lblempid.setFont(new Font("serif",Font.BOLD,20));**

**add(lblempid);**

**lblempID=new JLabel();**

**lblempID.setBounds(200,400,150,30);**

**lblempID.setFont(new Font("serif",Font.BOLD,20));**

**add(lblempID);**

**try{**

**cone c=new cone();**

**String query="select \* from employee where empId='"+empId+"'";**

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

**while(rs.next()){**

**lblname.setText(rs.getString("name"));**

**tffname.setText(rs.getString("fname"));**

**lbeldob.setText(rs.getString("dob"));**

**tfaddress.setText(rs.getString("address"));**

**tfsalary.setText(rs.getString("salary"));**

**tfphone.setText(rs.getString("phone"));**

**tfemail.setText(rs.getString("email"));**

**tfeducation.setText(rs.getString("education"));**

**tfdesignation.setText(rs.getString("designation"));**

**lbelaadhar.setText(rs.getString("aadhar"));**

**lblempID.setText(rs.getString("empId"));**

**}**

**}**

**catch(Exception e){**

**e.printStackTrace();**

**}**

**add= new JButton("Update Details");**

**add.setBounds(250,550,150,40);**

**add.setBackground(Color.BLACK);**

**add.setForeground(Color.WHITE);**

**add.addActionListener(this);**

**add(add);**

**back= new JButton("BACK");**

**back.setBounds(450,550,150,40);**

**back.setBackground(Color.BLACK);**

**back.setForeground(Color.WHITE);**

**back.addActionListener(this);**

**add(back);**

**ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("image/update.jpg"));**

**Image i2=i1.getImage().getScaledInstance(1110,700,Image.SCALE\_DEFAULT);**

**ImageIcon i3=new ImageIcon(i2);**

**JLabel image=new JLabel(i3);**

**image.setBounds(0,0,1120,630);**

**add(image);**

**setSize(900,700);**

**setLocation(300,50);**

**setVisible(true);**

**}**

**public void actionPerformed(ActionEvent ae){**

**if (ae.getSource()==add){**

**String fname=tffname.getText();**

**String salary=tfsalary.getText();**

**String address=tfaddress.getText();**

**String phone=tfphone.getText();**

**String email=tfemail.getText();**

**String education=(String)tfeducation.getText();**

**String designation=tfdesignation.getText();**

**try{**

**cone conn =new cone();**

**String query="update employee set fname='"+fname+"',salary='"+salary+"',address='"+address+"',phone='"+phone+"',email='"+email+"',education='"+education+"',designation='"+designation+"' where empId='"+empId+"'";**

**conn.s.executeUpdate(query);**

**JOptionPane.showMessageDialog(null,"Details updated successfully");**

**setVisible(false);**

**new Home();**

**}catch(Exception e){**

**e.printStackTrace();**

**}**

**}else**

**{**

**setVisible(false);**

**new Home();**

**}**

**}**

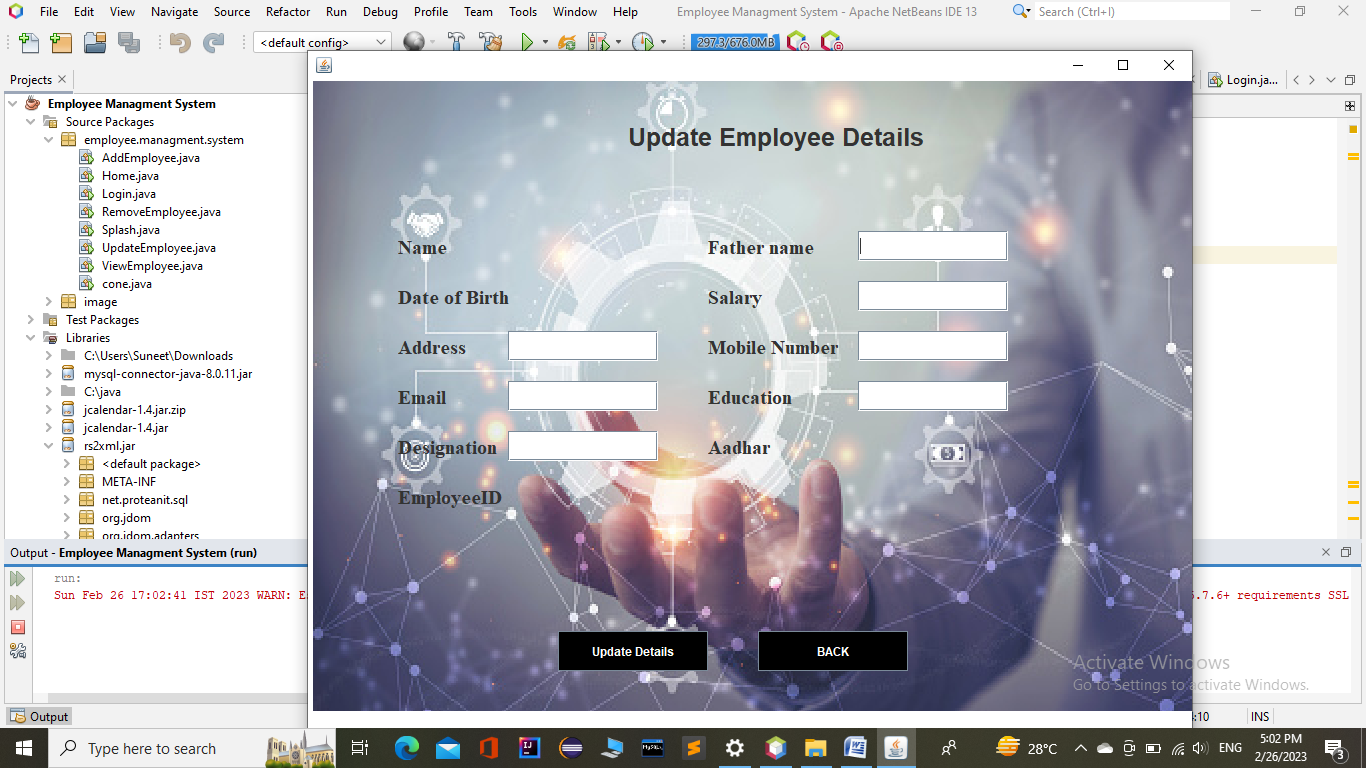
**public static void main (String args[])**

**{**

**new UpdateEmployee("");**

**}**

**}**



**Fig.6**

**Code of Start Page –Splash.java**

**package employee.managment.system;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**public class Splash extends JFrame implements ActionListener {**

**Splash(){**

**getContentPane().setBackground(Color.WHITE);**

**setLayout(null);**

**JLabel heading=new JLabel("IT Office Employees Record System");**

**heading.setBounds(80,30 ,1200, 60);**

**heading.setFont(new Font("serif",Font.BOLD,40));**

**heading.setForeground(Color.BLACK);**

**ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("image/empimg.jpg"));**

**Image i2=i1.getImage().getScaledInstance(1110,700,Image.SCALE\_DEFAULT);**

**ImageIcon i3=new ImageIcon(i2);**

**JLabel image=new JLabel(i3);**

**image.setBounds(20,100,1050,500);**

**add(image);**

**JButton clickhere=new JButton ("CLICK HERE TO CONTINUE");**

**clickhere .setBounds(350,350,300,70);**

**clickhere.setBackground(Color.BLACK);**

**clickhere.setForeground(Color.WHITE);**

**clickhere.addActionListener(this);**

**image.add(clickhere);**

**add(heading);**

**setSize(1110,700);**

**setLocation(150,50);**

**setVisible(true);**

**while (true)**

**{**

**heading.setVisible(false);**

**try{**

**Thread.sleep(500);**

**}catch(Exception e)**

**{**

**}**

**heading.setVisible(true);**

**try{**

**Thread.sleep(500);**

**}catch(Exception e){**

**}**

**}**

**}**

**public void actionPerformed(ActionEvent ae){**

**setVisible(false);**

**new Login();**

**}**

**public static void main (String args[])**

**{**

**new Splash();**

**}**

**}**

****

**Code For Create Connection Database Using MYSQL**

**package employee.managment.system;**

**import java.sql.\*;**

**public class cone {**

**Connection c;**

**Statement s;**

**public cone(){**

**try{**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**c=DriverManager.getConnection("jdbc:mysql://localhost:3306/EmployeeManagementSystem","root","Suneet@123");**

**s=c.createStatement();**

**Connection con =DriverManager.getConnection(url,username,password);**

**}catch(Exception e){**

**e.printStackTrace();**

**}**

**}**

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

**new cone();**

**}**

}

# 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 |

1. **Conclusion**

The goal of the initiative is to digitise personnel databases in businesses and provide administrators 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

The GUI and the features added to this system are the basic ones. 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.

# Bibliography

* 1. Basic Knowledge from Internshala Online Training Platform.
  2. I had take help from www.Youtub. vedios
  3. Julie Bulmash, “Human Resource Management and Technology”, Chapter 3.
  4. Ian Sommerville, “Software Engineering”, 9th Edition, Addison-Wesley, 2011.
  5. Avison, D. and Fitzgerald, G. (2003).Information systems Development Methodologies, Techniques and Tools.3rd Edition. McGraw-Hill Education Limited Bershire
  6. Juan Manuel Munoz Palacio, Information systems development methodologies for Data-driven Decision Support Systems, 2010.
  7. Renae Broderick, John W. Boudreau, “Human resource management, information technology, and the competitive edge”, Academy of Management Executive, 1992 Vol. 6 No.
  8. Deitel, PJ & Deitel, HM, 2008, Internet & World Wide Web How To Program,Dorling Kindersley, India.