# [Employee Management System]

## A MINI PROJECT REPORT

***Submitted by***

## [SADAB HUSSAIN]

## [92201703107]

## [MD ALIZAD]

## [92201703127]

## BACHELOR OF TECHNOLOGY

***in***

### Computer Engineering

****

## Marwadi University, Rajkot

**[OCTOBER,2022-23]**

****

## Marwadi University

**Rajkot**

# CERTIFICATE

# 

This is to certify that the project report submitted along with the project entitled **<EMPLOYEE MANAGEMENT SYSTEM>** has been carried out by **<SADAB HUSSAIN and MD ALIZAD, 92201703107 and 92201703127>** under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 3rd Semester of Marwadi University, Rajkot during the academic year 2022-23.

<Sign> <Sign>

<CHARMI VORA> <Dr. KRUNAL VAGHELA> Internal Guide Head of the Department

Computer Engineering Computer Engineering

Marwadi University Marwadi University

### 

## Marwadi University

**Rajkot**

# DECLARATION

We hereby declare that the Mini Project report submitted along with the Project entitled

**<EMPLOYEE MANAGEMENT SYSTEM>** submitted in partial fulfilment for the degree of Bachelor of Technology in <Computer Engineering> to Marwadi University, Rajkot, is a bonafide record of original project work carried out by me / us at Marwadi University under the supervision of **< CHARMI VORA>** and that no part of this report has been directly copied from any students’ reports or taken from any other source, without providing due reference.

Name of the Student Sign of Student

1. SADAB HUSSAIN
2. MD ALIZAD

### List of Figures

### 

Fig 1.1 Use case / Procedure Diagram ........................................................................ 2

Fig 1.2 Activity / Process Diagram ............................................................................. 5

Fig 2.1 Use case / Procedure Diagram ...................................................................... 12

Fig 2.2 Activity / Process Diagram ........................................................................... 15

Fig 2.3 Use case / Procedure Diagram ...................................................................... 22

Fig 2.4 Activity / Process Diagram ........................................................................... 25

Fig 3.1 Use case / Procedure Diagram ...................................................................... 32

Fig 4.1 Activity / Process Diagram ........................................................................... 35

Fig 4.2 Use case / Procedure Diagram ...................................................................... 42

Fig 4.3 Activity / Process Diagram ........................................................................... 43

### List of Tables

Table 1.1 Popular Methods / Techniques ................................................................... 2

Table 1.2 User / Reading / Observation Table ............................................................ 5

Table 2.1 Popular Methods / Techniques .................................................................. 12

Table 2.2 User / Reading / Observation Table .......................................................... 15

Table 2.3 Popular Methods / Techniques .................................................................. 22

Table 2.4 User / Reading / Observation Table .......................................................... 25

Table 3.1 Popular Methods / Techniques .................................................................. 32

Table 4.1 User / Reading / Observation Table .......................................................... 35

Table 4.2 Popular Methods / Techniques .................................................................. 42

Table 4.3 User / Reading / Observation Table .......................................................... 43

|  |  |
| --- | --- |
|  | **Abbreviations** |
| ALU | Arithmetical & Logical Unit |
| SDLC | Software Development Life Cycle |
|  |  |
|  |  |
|  |  |

### Table of Contents

Acknowledgement................................................................................................................... i

Abstract .................................................................................................................................. ii

List of Figures ...................................................................................................................... iii

List of Tables ........................................................................................................................ iv

List of Abbreviations ............................................................................................................. v Table of Contents .................................................................................................................. vi

Chapter 1 ………………....................................................................................................... 1

1.1 Introduction to Java ......................................................................................................... 2

1.1.1 Benefits of Java.................................................................................................... 3

1.2 Introduction to Swing/Servlet/JSP (depends on your project)......................................... 4

Chapter 2 ………………........................................................................................................ 8

2.1 Introduction to Project Topic............................................................................................ 9

2.1.1 How to do ......................................................................................................... 10

2.2 Drawbacks in Existing System ..................................................................................... 11 2.3 Advantages of Proposed System ................................................................................... 12

2.4 Functional Requirements…............................................................................................ 13

2.4.1 Tools .................................................................................................................. 14

2.4.2 Front End and Back End ................................................................................... 14

Chapter 3 ………………..................................................................................................... 15

3.1 Source code ………....................................................................................................... 14

Chapter 4 ………………..................................................................................................... 15

4.1 Screenshots ………........................................................................................................ 17

Chapter 5 ………………..................................................................................................... 15

4.1 Conclusion ………........................................................................................................ 17

4.2 Future Enhancement...................................................................................................... 17

References............................................................................................................................ 40

**CHAPTER 1**

**OVERVIEW OF JAVA**

**1.1 Introduction of Java**

Java is a popular, versatile, and widely-used programming language that was developed by Sun Microsystems (now owned by Oracle Corporation). It was first released in 1995 and has since become a cornerstone in software development. Java's popularity is attributed to its platform independence, robustness, and extensive libraries.  
**Features of Java**

**1. Platform Independence**

Java programs are compiled into an intermediate bytecode, which can run on any platform with a Java Virtual Machine (JVM). This "Write Once, Run Anywhere" feature makes Java highly portable.

**2. Object-Oriented**

Java is a pure object-oriented programming language, emphasizing the creation and manipulation of objects. This leads to modular and maintainable code.

**3. Simple and Familiar**

Java's syntax is similar to other C-based languages, making it relatively easy for developers to learn and use.

**4. Robust and Secure**

Java has strong memory management and exception handling, reducing the risk of crashes and security vulnerabilities. It includes a variety of security features like bytecode verification.

**5. Rich Standard Library**

Java offers an extensive library of classes and packages, covering everything from data structures to network communication. This library simplifies many common programming tasks.

**6. Multithreading**

Java supports multithreading, allowing programs to execute multiple tasks simultaneously. This is crucial for building responsive and efficient applications.

**7. High Performance**

With Just-In-Time (JIT) compilation and various optimizations, Java applications can achieve high performance, comparable to natively compiled languages.

**8. Garbage Collection**

Java automatically manages memory through garbage collection. Developers don't need to explicitly free memory, reducing the risk of memory leaks.

**9. Networking and RMI**

Java's robust networking capabilities and Remote Method Invocation (RMI) support enable the creation of distributed, networked applications.

**10. Dynamic and Extensible**

Java supports dynamic loading of classes, enabling the development of dynamic and extensible applications.

**CHAPTER 2**

**OVERVIEW OF PROJECT**

**2.1 Introduction of LMS**

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 "EMPLOYEE MANAGEMENT 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.

**Login Form(JFrame):**

A login form is a user interface element that allows to access their accounts by providing authentication credentials, typically a username and password. Its primary purpose is to verify the identity of the user and grant access to their personalized content or services. The key components of a login form include:

**Username:** Users enter their unique username or email address.

**Password Field**: Users input their secret password.

**Login Button**: Clicking this button submits the entered information for authentication.

The login form provides security by ensuring that only authorized users can access their accounts and perform specific actions, such as making purchases, managing personal information, or accessing order history.

**HomePage(JFrame):**

A employee management system is used for adding new employee to a database, catalog, or online platform. It is an essential tool for add to new employee maintain a comprehensive list of the employee.

**Employee Name:** Name of new Employee**.**

**Employee Id:** ID of Employee.

**Employee Fname**: Employee Father’s Name.

**Date of birth:** Employee DOB.

**Salary:** Salary of Employee.

**Others Details:**

**Add Components:**

Inside the JFrame, add components like buttons, labels, text fields, and tables to facilitate CRUD operations. In your case, you'd need buttons for "Add," "Update," and "Delete," as well as buttons for at the Time of Adding Employee. You'll also need input fields for data entry and a table to display existing data.

**CRUD Operations:**

**Create (Add):** When the "Add" button is clicked, you can open a dialog or input fields for the user to enter data. This data can be added to the table displaying the records.

**Update (Edit):** When the "Update" button is clicked, allow the user to select a row in the table to edit, then update the data.

**Delete:** When the "Delete" button is clicked, remove the selected row from the table.

Export to Details:

When the "Export to Details" button is clicked, you can write the data from your table to a file. You might want to print the employee details like into the pdf.

**Export to PDF:**

For exporting to PDF, you can use a library like iText to create a PDF document. You'll need to format your data and generate a PDF document that includes the data from your table.

**User Interface Design:**

Pay attention to the design and layout of your JFrame to ensure a user-friendly and aesthetically pleasing interface.

**Testing and Debugging:**

Test your application thoroughly, handle exceptions, and debug any issues that may arise during testing.

Implementing the above steps will give you a basic framework for your Java Swing application that includes CRUD operations and data export to PDF. The specific code for your application would depend on your requirements and the libraries you choose to use.

Our project focuses on implementing an **Employee Management System** using Java, JDBC, JFrame, and Swing,GUI.

**2.2 Drawbacks of Existing System**

* Lack of Confirmation
* User Experience
* Export Options
* Data Validation
* Data Backup
* Scalability
* Concurrency
* Security
* Maintainability

**2.3 Advantages of Proposed System**

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

**2.4 Functional Requirements**

**1)** Java Development Kit (JDK) installed.

**2)** A database (e.g., MySQL) set up.

**3)** Java IDE (e.g., NetBeans) for development.

**4)** my sql-connector.jar file

**5)** jcalander.jar file

**6)** r2xml.jar

**2.4.1Tools** We employ the following tools for developing our Login and Product Registration

1. **Java**: A versatile, platform-independent programming language known for its security and robustness.
2. **JDBC (Java Database Connectivity):** JDBC enables database connectivity, facilitating data storage and retrieval.
3. **JFrame**: A Java GUI framework for creating windows, dialogs, and other graphical components.
4. **Swing**: Swing provides a set of user interface components for building a rich and interactive graphical interface.
5. **GUI**: A GUI, or Graphical User Interface, is a visual way for users to interact with electronic devices or software applications, using graphical elements like icons, buttons, windows, and menus to perform tasks and access functions, as opposed to text-based interfaces.

**2.4.2 Front End and Back End**

**1) Front End:** The front end of our **Employee Management System** is developed using Java's Swing and JFrame, ensuring a user-friendly and visually appealing interface. Students and instructors interact with the system through this interface.

**2) Back End:** The back end of the **Employee Management System** is powered by Java and JDBC, allowing us to securely store and manage user data, course content, and student progress. The back end handles database operations and ensures data integrity.

**CHAPTER 3**

**PROJECT SOURCE CODE**

**3.1 Employee Management System Source Code**

**Splash.java(Start From Here)**

package employee.management.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("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, 1050, 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);

setSize(1170, 650);

setLocation(200, 50);

setVisible(true);

setTitle("Employee Management System");

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();

}

}

**Login.java**

package employee.management.system;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

public class Login extends JFrame implements ActionListener{

JTextField tfusername, tfpassword;

Login() {

getContentPane().setBackground(Color.WHITE);

setLayout(null);

JLabel lblusername = new JLabel("Username");

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

add(lblusername);

tfusername = new JTextField();

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

add(tfusername);

JLabel lblpassword = new JLabel("Password");

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

add(lblpassword);

tfpassword = new JTextField();

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

add(tfpassword);

JButton login = new JButton("LOGIN");

login.setBounds(150, 140, 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, 200);

setVisible(true);

setTitle("Login");

}

public void actionPerformed(ActionEvent ae) {

try {

String username = tfusername.getText();

String password = tfpassword.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 Home();

} else {

JOptionPane.showMessageDialog(null, "Invalid username or password");

setVisible(false);

}

} catch (Exception e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

new Login();

}

}

**home.java**

package employee.management.system;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Home extends JFrame implements ActionListener{

JButton view, add, update, remove;

Home() {

setLayout(null);

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/home.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(620, 20, 400, 40);

heading.setFont(new Font("Raleway", 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);

setSize(1120, 630);

setLocation(250, 100);

setVisible(true);

setTitle("Home");

}

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 {

setVisible(false);

new RemoveEmployee();

}

}

public static void main(String[] args) {

new Home();

}

}

**AddEmployee.java**

package employee.management.system;

import java.awt.\*;

import javax.swing.\*;

import com.toedter.calendar.JDateChooser;

import java.util.\*;

import java.awt.event.\*;

public class AddEmployee extends JFrame implements ActionListener{

Random ran = new Random();

int number = ran.nextInt(999999);

JTextField tfname, tffname, tfaddress, tfphone, tfaadhar, tfemail, tfsalary, tfdesignation;

JDateChooser dcdob;

JComboBox cbeducation;

JLabel lblempId;

JButton add, back;

AddEmployee() {

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 labelname = new JLabel("Name");

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

labelname.setFont(new Font("serif", Font.PLAIN, 20));

add(labelname);

tfname = new JTextField();

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

add(tfname);

JLabel labelfname = new JLabel("Father's Name");

labelfname.setBounds(400, 150, 150, 30);

labelfname.setFont(new Font("serif", Font.PLAIN, 20));

add(labelfname);

tffname = new JTextField();

tffname.setBounds(600, 150, 150, 30);

add(tffname);

JLabel labeldob = new JLabel("Date of Birth");

labeldob.setBounds(50, 200, 150, 30);

labeldob.setFont(new Font("serif", Font.PLAIN, 20));

add(labeldob);

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.PLAIN, 20));

add(labelsalary);

tfsalary = new JTextField();

tfsalary.setBounds(600, 200, 150, 30);

add(tfsalary);

JLabel labeladdress = new JLabel("Address");

labeladdress.setBounds(50, 250, 150, 30);

labeladdress.setFont(new Font("serif", Font.PLAIN, 20));

add(labeladdress);

tfaddress = new JTextField();

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

add(tfaddress);

JLabel labelphone = new JLabel("Phone");

labelphone.setBounds(400, 250, 150, 30);

labelphone.setFont(new Font("serif", Font.PLAIN, 20));

add(labelphone);

tfphone = new JTextField();

tfphone.setBounds(600, 250, 150, 30);

add(tfphone);

JLabel labelemail = new JLabel("Email");

labelemail.setBounds(50, 300, 150, 30);

labelemail.setFont(new Font("serif", Font.PLAIN, 20));

add(labelemail);

tfemail = new JTextField();

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

add(tfemail);

JLabel labeleducation = new JLabel("Higest Education");

labeleducation.setBounds(400, 300, 150, 30);

labeleducation.setFont(new Font("serif", Font.PLAIN, 20));

add(labeleducation);

String courses[] = {"BBA", "BCA", "BA", "BSC", "B.COM", "BTech", "MBA", "MCA", "MA", "MTech", "MSC", "PHD"};

cbeducation = new JComboBox(courses);

cbeducation.setBackground(Color.WHITE);

cbeducation.setBounds(600, 300, 150, 30);

add(cbeducation);

JLabel labeldesignation = new JLabel("Designation");

labeldesignation.setBounds(50, 350, 150, 30);

labeldesignation.setFont(new Font("serif", Font.PLAIN, 20));

add(labeldesignation);

tfdesignation = new JTextField();

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

add(tfdesignation);

JLabel labelaadhar = new JLabel("Aadhar Number");

labelaadhar.setBounds(400, 350, 150, 30);

labelaadhar.setFont(new Font("serif", Font.PLAIN, 20));

add(labelaadhar);

tfaadhar = new JTextField();

tfaadhar.setBounds(600, 350, 150, 30);

add(tfaadhar);

JLabel labelempId = new JLabel("Employee id");

labelempId.setBounds(50, 400, 150, 30);

labelempId.setFont(new Font("serif", Font.PLAIN, 20));

add(labelempId);

lblempId = new JLabel("" + number);

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

lblempId.setFont(new Font("serif", Font.PLAIN, 20));

add(lblempId);

add = new JButton("Add Details");

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

add.addActionListener(this);

add.setBackground(Color.BLACK);

add.setForeground(Color.WHITE);

add(add);

back = new JButton("Back");

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

back.addActionListener(this);

back.setBackground(Color.BLACK);

back.setForeground(Color.WHITE);

add(back);

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/add\_employee.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);

setSize(900, 700);

setLocation(300, 50);

setVisible(true);

setTitle("Add Employee");

}

public void actionPerformed(ActionEvent ae) {

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

String name = tfname.getText();

String fname = tffname.getText();

String dob = ((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 {

Conn conn = new Conn();

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();

}

}

**ViewEmployee.java**

package employee.management.system;

import javax.swing.\*;

import java.awt.\*;

import java.sql.\*;

import net.proteanit.sql.DbUtils;

import java.awt.event.\*;

public class ViewEmployee extends JFrame implements ActionListener{

JTable table;

Choice cemployeeId;

JButton search, print, update, back;

ViewEmployee() {

getContentPane().setBackground(Color.WHITE);

setLayout(null);

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

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

add(searchlbl);

cemployeeId = new Choice();

cemployeeId.setBounds(180, 20, 150, 20);

add(cemployeeId);

try {

Conn c = new Conn();

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 {

Conn c = new Conn();

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

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

} catch (Exception e) {

e.printStackTrace();

}

JScrollPane jsp = new JScrollPane(table);

jsp.setBounds(0, 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);

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

setSize(900, 700);

setLocation(300, 100);

setVisible(true);

setTitle("View Employee");

}

public void actionPerformed(ActionEvent ae) {

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

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

}

}

**UpdatedEmployee.java**

package employee.management.system;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import java.sql.\*;

public class UpdateEmployee extends JFrame implements ActionListener{

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

JLabel lblempId;

JButton add, back;

String empId;

UpdateEmployee(String empId) {

this.empId = empId;

getContentPane().setBackground(Color.WHITE);

setLayout(null);

JLabel heading = new JLabel("Update Employee Detail");

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

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

add(heading);

tffname = new JTextField();

tffname.setBounds(600, 150, 150, 30);

add(tffname);

JLabel labelname = new JLabel("Name");

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

labelname.setFont(new Font("serif", Font.PLAIN, 20));

add(labelname);

JLabel lblname = new JLabel();

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

add(lblname);

JLabel labelfname = new JLabel("Father's Name");

labelfname.setBounds(400, 150, 150, 30);

labelfname.setFont(new Font("serif", Font.PLAIN, 20));

add(labelfname);

JLabel labeldob = new JLabel("Date of Birth");

labeldob.setBounds(50, 200, 150, 30);

labeldob.setFont(new Font("serif", Font.PLAIN, 20));

add(labeldob);

JLabel lbldob = new JLabel();

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

add(lbldob);

JLabel labelsalary = new JLabel("Salary");

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

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

add(labelsalary);

tfsalary = new JTextField();

tfsalary.setBounds(600, 200, 150, 30);

add(tfsalary);

JLabel labeladdress = new JLabel("Address");

labeladdress.setBounds(50, 250, 150, 30);

labeladdress.setFont(new Font("serif", Font.PLAIN, 20));

add(labeladdress);

tfaddress = new JTextField();

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

add(tfaddress);

JLabel labelphone = new JLabel("Phone");

labelphone.setBounds(400, 250, 150, 30);

labelphone.setFont(new Font("serif", Font.PLAIN, 20));

add(labelphone);

tfphone = new JTextField();

tfphone.setBounds(600, 250, 150, 30);

add(tfphone);

JLabel labelemail = new JLabel("Email");

labelemail.setBounds(50, 300, 150, 30);

labelemail.setFont(new Font("serif", Font.PLAIN, 20));

add(labelemail);

tfemail = new JTextField();

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

add(tfemail);

JLabel labeleducation = new JLabel("Higest Education");

labeleducation.setBounds(400, 300, 150, 30);

labeleducation.setFont(new Font("serif", Font.PLAIN, 20));

add(labeleducation);

tfeducation = new JTextField();

tfeducation.setBounds(600, 300, 150, 30);

add(tfeducation);

JLabel labeldesignation = new JLabel("Designation");

labeldesignation.setBounds(50, 350, 150, 30);

labeldesignation.setFont(new Font("serif", Font.PLAIN, 20));

add(labeldesignation);

tfdesignation = new JTextField();

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

add(tfdesignation);

JLabel labelaadhar = new JLabel("Aadhar Number");

labelaadhar.setBounds(400, 350, 150, 30);

labelaadhar.setFont(new Font("serif", Font.PLAIN, 20));

add(labelaadhar);

JLabel lblaadhar = new JLabel();

lblaadhar.setBounds(600, 350, 150, 30);

add(lblaadhar);

JLabel labelempId = new JLabel("Employee id");

labelempId.setBounds(50, 400, 150, 30);

labelempId.setFont(new Font("serif", Font.PLAIN, 20));

add(labelempId);

lblempId = new JLabel();

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

lblempId.setFont(new Font("serif", Font.PLAIN, 20));

add(lblempId);

try {

Conn c = new Conn();

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"));

lbldob.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"));

lblaadhar.setText(rs.getString("aadhar"));

lblempId.setText(rs.getString("empId"));

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

}

} catch (Exception e) {

e.printStackTrace();

}

add = new JButton("Update Details");

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

add.addActionListener(this);

add.setBackground(Color.BLACK);

add.setForeground(Color.WHITE);

add(add);

back = new JButton("Back");

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

back.addActionListener(this);

back.setBackground(Color.BLACK);

back.setForeground(Color.WHITE);

add(back);

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

setSize(900, 700);

setLocation(300, 50);

setVisible(true);

setTitle("Update Employee");

}

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 = tfeducation.getText();

String designation = tfdesignation.getText();

try {

Conn conn = new Conn();

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("");

}

}

**RemoveEmployee.java**

package employee.management.system;

import javax.swing.\*;

import java.awt.\*;

import java.sql.\*;

import java.awt.event.\*;

public class RemoveEmployee extends JFrame implements ActionListener {

Choice cEmpId;

JButton delete, back;

RemoveEmployee() {

getContentPane().setBackground(Color.WHITE);

setLayout(null);

JLabel labelempId = new JLabel("Employee Id");

labelempId.setBounds(50, 50, 100, 30);

add(labelempId);

cEmpId = new Choice();

cEmpId.setBounds(200, 50, 150, 30);

add(cEmpId);

try {

Conn c = new Conn();

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 labelphone = new JLabel("Phone");

labelphone.setBounds(50, 150, 100, 30);

add(labelphone);

JLabel lblphone = new JLabel();

lblphone.setBounds(200, 150, 100, 30);

add(lblphone);

JLabel labelemail = new JLabel("Email");

labelemail.setBounds(50, 200, 100, 30);

add(labelemail);

JLabel lblemail = new JLabel();

lblemail.setBounds(200, 200, 100, 30);

add(lblemail);

try {

Conn c = new Conn();

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

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

while(rs.next()) {

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

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

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

}

} catch (Exception e) {

e.printStackTrace();

}

cEmpId.addItemListener(new ItemListener() {

public void itemStateChanged(ItemEvent ie) {

try {

Conn c = new Conn();

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

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

while(rs.next()) {

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

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

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

}

} catch (Exception e) {

e.printStackTrace();

}

}

});

delete = new JButton("Delete");

delete.setBounds(80, 300, 100,30);

delete.setBackground(Color.BLACK);

delete.setForeground(Color.WHITE);

delete.addActionListener(this);

add(delete);

back = new JButton("Back");

back.setBounds(220, 300, 100,30);

back.setBackground(Color.BLACK);

back.setForeground(Color.WHITE);

back.addActionListener(this);

add(back);

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(350, 0, 600, 400);

add(image);

setSize(1000, 400);

setLocation(300, 150);

setVisible(true);

setTitle("Remove Employee");

}

public void actionPerformed(ActionEvent ae) {

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

try {

Conn c = new Conn();

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

c.s.executeUpdate(query);

JOptionPane.showMessageDialog(null, "Employee Information Deleted Sucessfully");

setVisible(false);

new Home();

} catch (Exception e) {

e.printStackTrace();

}

} else {

setVisible(false);

new Home();

}

}

public static void main(String[] args) {

new RemoveEmployee();

}

}

**Conn.java**

package employee.management.system;

import java.sql.\*;

public class Conn {

Connection c;

Statement s;

public Conn() {

try {

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

c = DriverManager.getConnection("jdbc:mysql:/// employeemanagementsystem", "root", "");

s = c.createStatement();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**CHAPTER 4**

**SCREENSHOTS**

**4.1 Admin Screenshot**

**(minimum 5 screenshots)**

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.

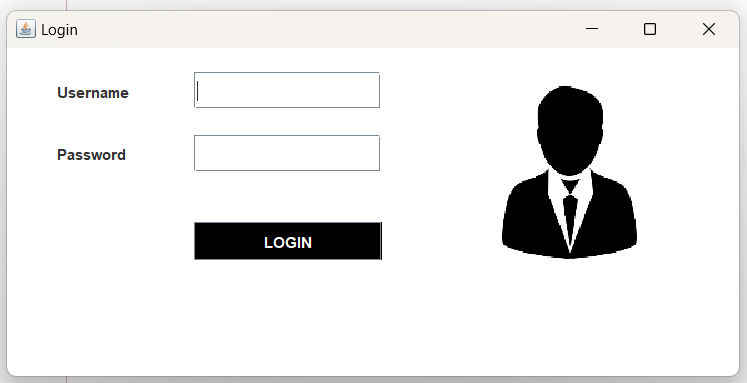


Fig. 1

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 view Employees Update Employee.



Fig.2

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

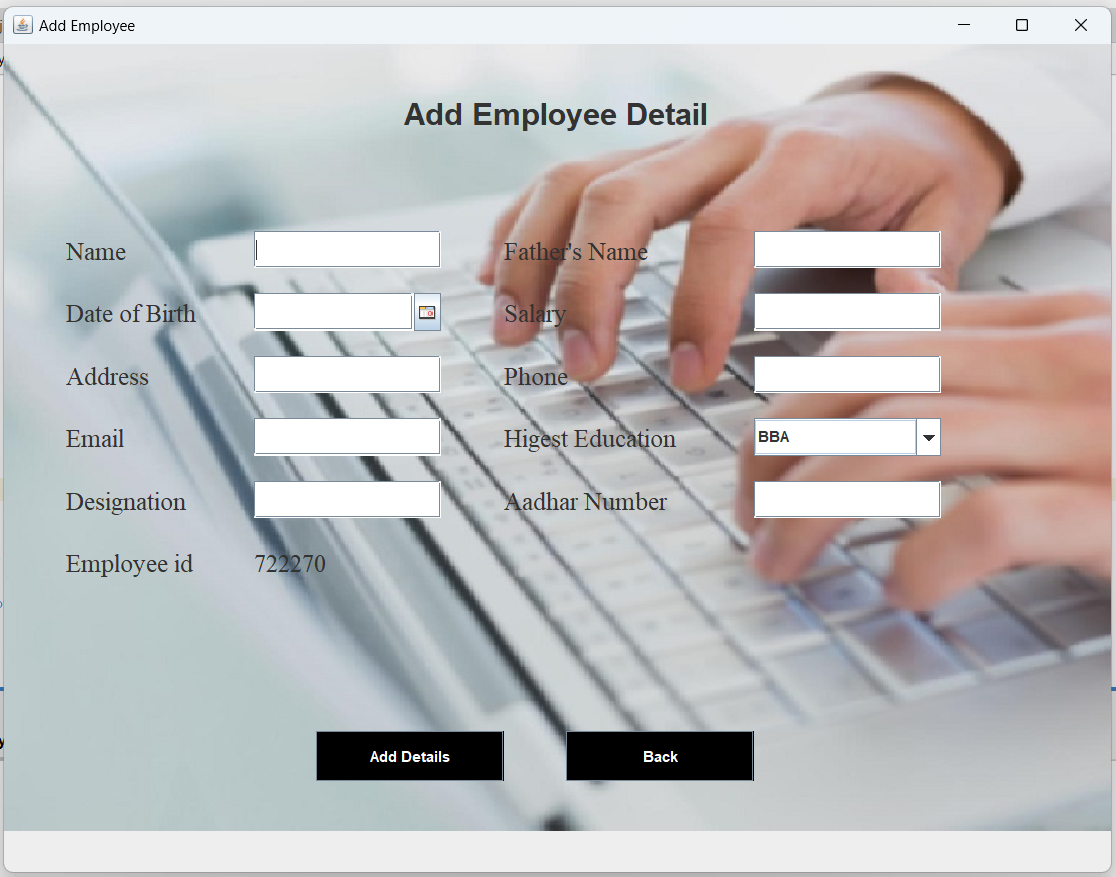


Fig.3

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

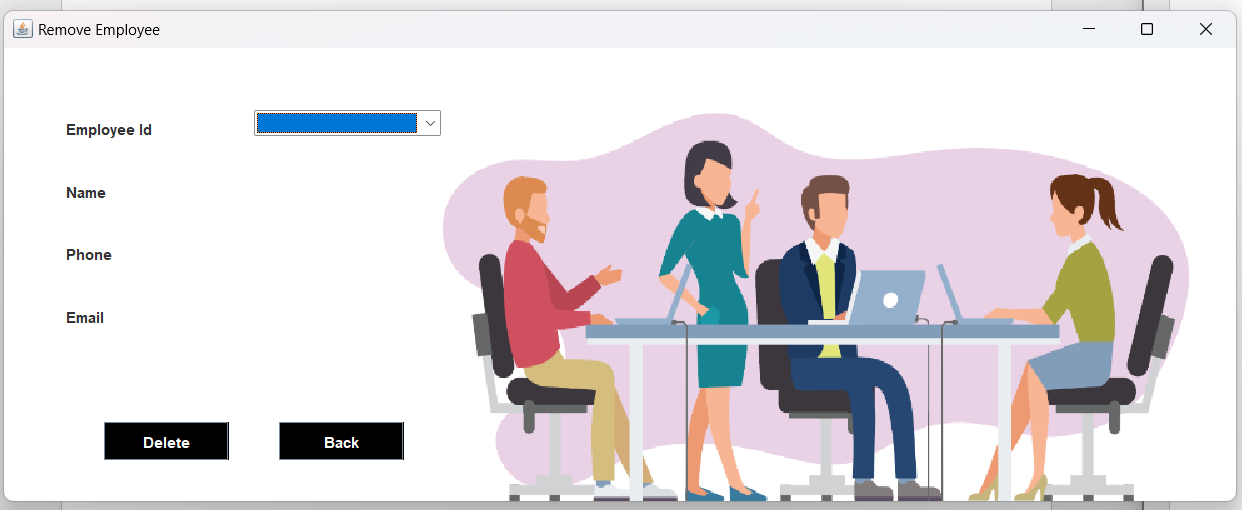


Fig.4

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

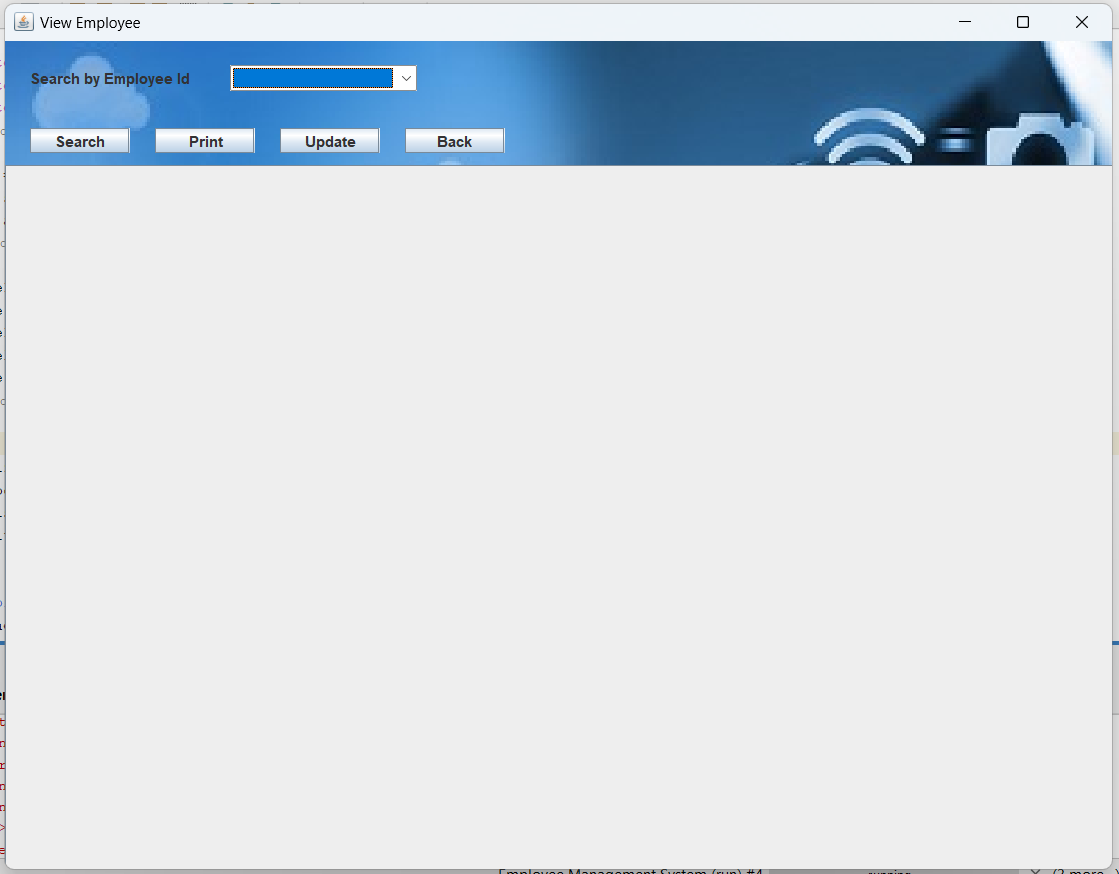


Fig.5

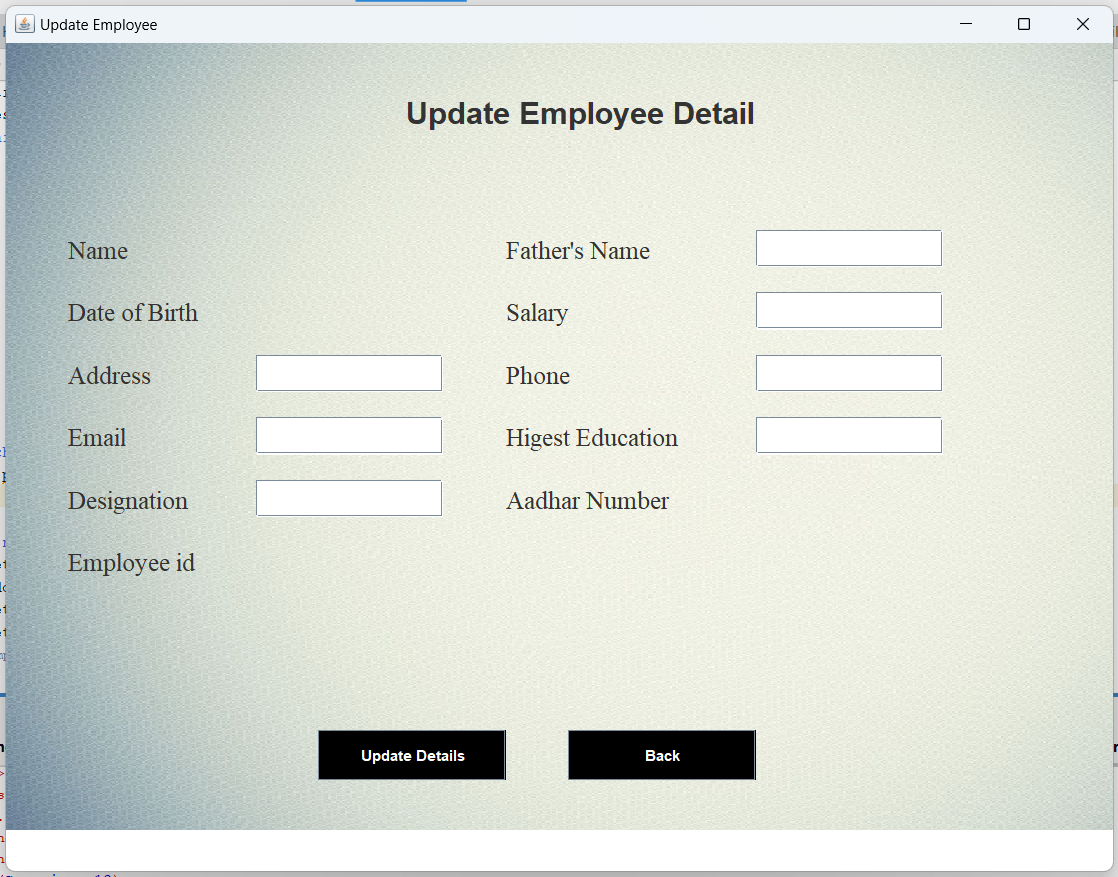


Fig.6

**CHAPTER 5**

**CONCLUSION AND FUTURE ENHANCEMENTS**

**5.1 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.

**5.2 Future Enhancements**

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.

**REFERENCES**

**<In IEEE Format – Add minimum 5 references>**

[1] Axelrod, CW (2013). Managing the Risks of Cyber-Physical Systems. In 2013 Systems,

Applications and Technology Conf. (LISAT), pp. 1–6. IEEE: Long Island.

[2] www.researchgate.com/doc/library\_management.

[3] Renae Broderick, John W. Boudreau, “Human resource management, information technology, and the competitive edge”, Academy of Management Executive, 1992 Vol. 6 No. 2.

[4] Avison, D. and Fitzgerald, G. (2003).Information systems Development Methodologies, Techniques and Tools.3rd Edition. McGraw-Hill Education Limited Bershire.

[5] Deitel, PJ & Deitel, HM, 2008, Internet & World Wide Web How to Program, Dorling Kindersley, India.