VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi – 590 018

Advanced Java Laboratory with Mini project (20CSEL57)

Report on

"Employee Management System"

Submitted By

Sujith S R	**1GA20CS118**
Prateek Patil	**1GA20CS101**

Under the Guidance of

Mrs. Haripriya C
Assistant Professor
Dept. of CSE

Department of Computer Science and Engineering
(Accredited by NBA 2022-2025)
GLOBAL ACADEMY OF TECHNOLOGY
Rajarajeshwarinagar, Bengaluru – 560 098

2022 - 2023

GLOBAL ACADEMY OF TECHNOLOGY

An Autonomous Institute, Affiliated to VTU Belagavi,
Approved by AICTE, Accredited by NAAC,'A' Grade,
Ideal Homes Township, Rajarajeshwari Nagar, Bengaluru – 560 098

Department of Computer Science and Engineering

(Accredited by NBA 2022-2025)

CERTIFICATE

Certified that the V Semester Advanced Java Laboratory with Mini project entitled **"Employee Management System"** carried out by **Sujith S R [1GA20CS118], Prateek Patil [1GA20CS101]** are bonafide students of Global Academy of Technology, Bachelor of Engineering in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi during the year 2022-2023. It is certified that all the corrections/suggestions indicated have been incorporated in the report submitted.

Dr. K Anantha Padmanabha Professor Dept. of CSE GAT, Bengaluru	Mrs. Haripriya C Assistant Professor Dept. of CSE GAT, Bengaluru	Dr. Kumaraswamy S Professor and HOD Dept. of CSE GAT, Bengaluru

Name and Signature of the Students

Sujith S R

Prateek Patil

Signature of Internal Examiner **Signature of External Examiner**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance crowned our efforts with success.

We consider ourselves proud, to be part of **Global Academy of Technology** family, the Institution which stood by our way in endeavors.

We express our deep and sincere thanks to our Principal **Dr. N. Rana Pratap Reddy** for his support.

We are grateful to **Dr. Kumaraswamy S,** Professor and HOD, Dept. of CSE for his inspiration and invaluable help in channelizing our efforts in right direction.

We wish to thank our internal guides **Dr. K Anantha Padmanabha,** Professor and **Mrs. V**, Assistant Professor, Dept. of CSE for guiding and correcting various documents of ours with attention and care. They have taken initiative to go through the document and make necessary corrections as and when needed.

We would like to thank the faculty members and supporting staff of the Department of CSE, GAT for providing all the support for completing the Project work.

Finally, we are grateful to our parents and friends for their unconditional support and help during our Project work.

Student Names

Sujith S R **[1GA20CS118]**

Prateek **[1GA20CS101]**

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.

Everything has been digitized 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.

TABLE OF CONTENTS

List of Figures

PROJECT REPORT OUTLINE

1 CHAPTER 1:

A brief description of Java highlighting its features.

2 CHAPTER 2:

System Requirements Specifications:

The Description of required Hardware and software for running the project application.

3 CHAPTER 3:

The Program Code.

4 CHAPTER 4:

Snapshots displaying outputs in each input case.

CHAPTER 1

INTRODUCTION

1.1 Introduction to Advanced Java Concepts

1.1.1 Collections

The Collection in Java is a framework that provides an architecture to store and manipulate the group of objects.

Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).

Before the Collection Framework(or before JDK 1.2) was introduced, the standard methods for grouping Java objects (or collections) were Arrays or Vectors, or Hashtables. All these collections had no common interface. Therefore, though the main aim of all the collections is the same, the implementation of all these collections was defined independently and had no correlation among them. And also, it is very difficult for the users to remember all the different methods, syntax, and constructors present in every collection class.

1.1.2 Servlets

- Servlets are small programs that execute on the server side of a web connection.
- Applets dynamically extend the functionality of a web browser where as servlets dynamically extend the functionality of a web server.
- Servlets are the Java programs that run on the Java-enabled web server or application server.
- They are used to handle the request obtained from the web server, process the request, produce the response, then send response back to the web server.

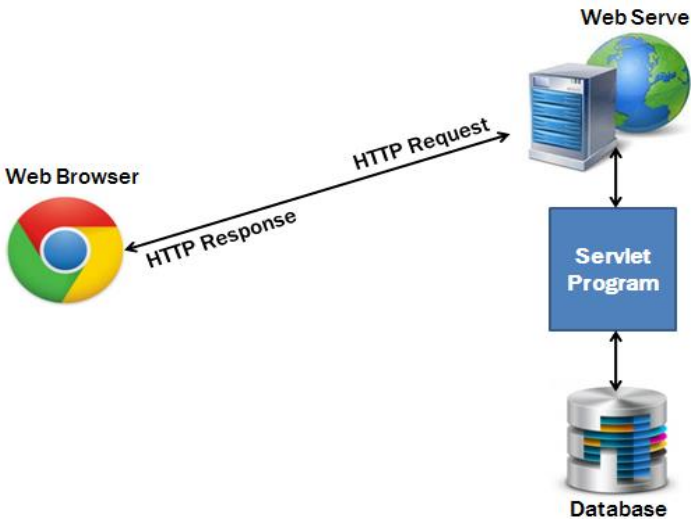

Fig 1.1 Working of Servlet

Execution of Servlet

- The clients send the request to the web server.
- The web server receives the request.
- The web server passes the request to the corresponding servlet.
- The servlet processes the request and generates the response in the form of output.
- The servlet sends the response back to the web server.
- The web server sends the response back to the client and the client browser displays it on the screen.

1.1.3 JSP

Java Server Pages (JSP) technology provides a simplified, fast way to create dynamic web content. Java Server Pages (JSP) technology enables Web developers and designers to rapidly develop and easily maintain, information, dynamic Web pages that leverage existing business systems. JSP technology uses HTML/XML-like tags that encapsulate the logic that generates the content for the page. A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

1.1.4 JDBC

JDBC or Java Database Connectivity is a Java API to connect and execute the query with the database. It is a specification from Sun microsystems that provides a standard abstraction (API or Protocol) for java applications to communicate with various databases. It provides the language with java database connectivity standards. It is used to write programs required to access databases. JDBC, along with the database driver, can access databases and spreadsheets. The enterprise data stored in a relational database can be accessed with the help of JDBC APIs.

JDBC is an API (Application programming interface) used in java programming to interact with databases. The classes and interfaces of JDBC allow the application to send requests made by users to the specified database.

CHAPTER 2

SYSTEM DEFINITION

2.1 PROBLEM STATEMENT AND OBJECTIVES

The current manual system of managing employee records is often time-consuming and prone to errors, leading to decreased productivity and efficiency. In order to overcome these challenges, organizations require an automated system that can manage employee data in a more efficient and organized manner. Therefore, the objective of this project is to design and develop an Employee Management System using Java programming language that can address these challenges and help organizations manage their employee data in a better way. The system should have a user-friendly interface that allows administrators to manage employee information easily and efficiently. Additionally, the system should have role-based access control, data security, and appropriate validation and verification mechanisms to ensure the integrity of employee data.

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.

2.2 SYSTEM REQUIREMENTS SPECIFICATION

SOFTWARE REQUIREMENTS:

- Operating system : Microsoft Windows 10
- IDE : Apache NetBeans IDE 13
- Programming Language : JAVA

HARDWARE REQUIREMENTS:

- Processor : Intel Core i5 or later versions
- Memory : 512 MB RAM
- 2GB Hard Disk Drive
- Mouse or other pointing device
- Keyboard
- Display device

MISCELLANEOUS REQUIREMENTS:

- All the required library files and the header files should be available in the include directory.

CHAPTER 3

IMPLEMENTATION

3.1 PROGRAM CODE

1.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);
    }


     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();
  }
}
```

2.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(100, 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 {
  setVisible(false);
  new RemoveEmployee();
  }
  }

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

3.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",
  "sujith");
           s = c.createStatement();
        } catch (Exception e) {
           e.printStackTrace();
        }
```

```
    }
}
```

4.AddEmployee.java

```
package employee.management.system;

import java.awt.*;
import javax.swing.*;
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() {
        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 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);

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

    @Override
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == add) {
            String name = tfname.getText();
            String fname = tffname.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();
    }

    private void add(JDateChooser dcdob) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from
  nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
}
```

CHAPTER 4

TESTING AND RESULTS

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing.

4.1 TESTING

1. **Unit Testing**

   Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components.

2. **Module Testing**

   A module is a collection of dependent components such as a object class, an abstract data type or some looser collection of procedures and functions. They are module related components, so can be tested without other system modules.

3. **System Testing**

   This is concerned with finding errors that result from unanticipated interaction between sub-system interface problems**.**

4. **Acceptance Testing**

   The system is tested with data supplied by the system customer rather than simulated test data.

4.2 SNAPSHOTS

Fig 4.2.1: Front Page

Fig 4.2.2: Login

Fig 4.2.3: Home Page

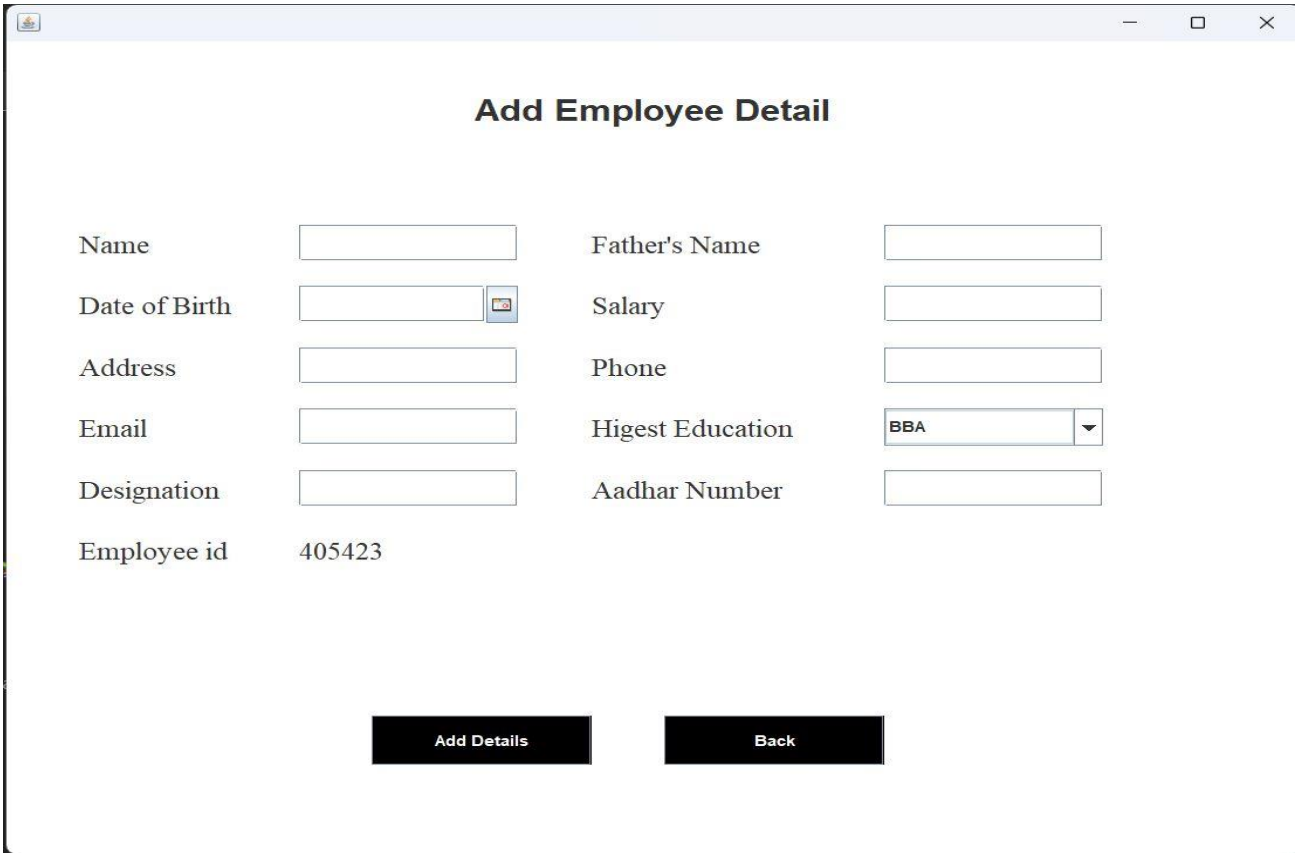

Fig 4.2.4: Add Employee Details

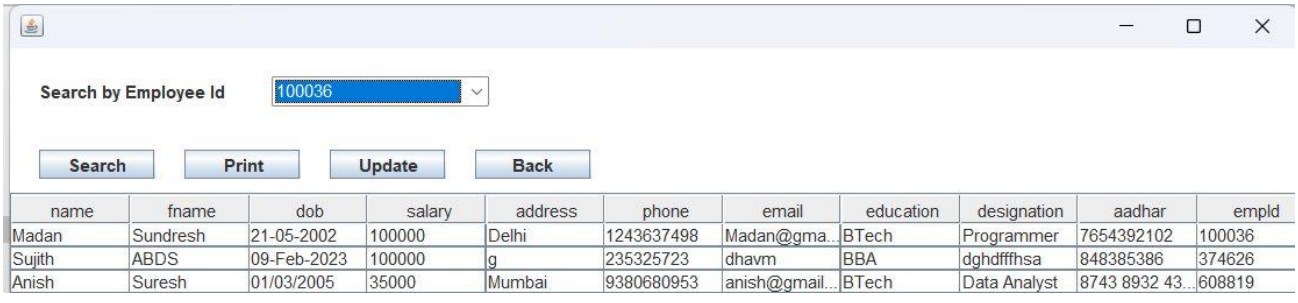

Fig 4.2.5: View Employee Details

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.

BIBLIOGRAPHY

[1] The Complete reference java seventh edition-Herbert Schildt

[2] https://www.stackoverflow.com

[3] www.javatpoint.com

[4] www.geeksforgeeks.com