**Part B Micro-Project Report**

**Employee Record Management System**

1. **Rationale:**

The Employee Record Management System powered by Java Swing is a practical solution for businesses seeking an efficient and user-friendly way to manage their employee records. By providing the ability to add, view, edit, and search employee details, our application empowers organizations to maintain accurate records and make informed HR decisions. With the constant evolution of this system, we aim to further enhance its functionality and contribute to improved workforce management practices.

1. **Course Outcomes Achieved:**
2. Develop program using GUI framework (AWT and Swing).
3. Handle events of AWT and Swing Components.
4. Develop programs using Database.
5. Develop programs to handle events in Java Programming
6. **Literature Review:**
   1. Employee Record Management System Using Java and MySQL by Deepak Saini and Nitesh Kumar (2020): This paper presents the design and implementation of an ERMS in Java that uses JavaFX for the user interface and MySQL for the database management system. The system includes features such as employee registration, attendance tracking, leave management, and salary processing. The system was tested, and the results showed that the system is efficient and user-friendly.
   2. Java-Based Human Resource Management System by R.N. Saranya, A. Santhoshkumar, and S. Saravanan (2019): This paper presents the design and implementation of a Java-based HRMS that includes modules for employee information management, attendance tracking, leave management, and payroll processing. The authors used Java Server Faces (JSF) and MySQL for the web-based interface and database management system, respectively. The system's performance was evaluated, and the results showed that the system is efficient and user-friendly.
   3. Employee Management System in Java by Amit Kumar (2018): This paper proposes an ERMS in Java that uses Java Server Pages (JSP) and MySQL for the web-based interface and database management system, respectively. The system includes features such as employee registration, attendance tracking, and leave management. The system's performance was evaluated, and the results showed that the system is scalable and efficient.
   4. Java Employee Record Management System (JERMS) by Anuja P. and Chavan S. (2016): This paper presents the design and implementation of JERMS, a Java-based ERMS. The system includes modules for managing employee details, salary information, attendance records, and performance evaluation. The authors used Java Swing for the user interface and MySQL for the database management system. The system was tested and evaluated, and the results showed that JERMS is a user-friendly and efficient ERMS.
7. **Actual Methodology Followed:**

* **setSize:** A method in Java used with graphical user interfaces (GUI) to set the dimensions (width and height) of a component, typically a window like a JFrame. It specifies the initial size of the component when it is displayed on the screen.
* **setLayout:** A method in Java used for setting the layout manager of a container, like a JPanel or JFrame, in a graphical user interface (GUI) application. The layout manager determines how components are arranged within the container. There are various layout managers, such as FlowLayout, BorderLayout, and GridLayout, each specifying a different approach to component positioning and sizing.
* **setBounds:** A method in Java that is used to set the position and size of a graphical component, like a JFrame or JButton. It takes four parameters: the x and y coordinates (position) and the width and height (size) of the component. This method allows you to precisely control the location and dimensions of a component on the GUI.
* setFocusPainted: a method used in Java to control whether a component, typically a button, should visually indicate focus. When you call setFocusPainted(true) on a component, like a JButton, it will display a focus border or some other visual indicator when it has keyboard focus, which means it is ready to accept input from the keyboard. When set to false, the component won't display this focus indicator. This can be useful for customizing the appearance of buttons and other components in a graphical user interface.
* setFont: a method in Java used to set the font (including typeface, size, and style) for a graphical component, such as a label, button, or text field, in a graphical user interface (GUI). By calling setFont, you can control the appearance of the text within the component, allowing you to change its size, style (e.g., bold or italic), and typeface (e.g., Arial, Times New Roman, etc.).
* add: a method in Java that's commonly used to add graphical components (like buttons, labels, and text fields) to a container (such as a JPanel, JFrame, or other GUI component). It allows you to place and arrange these components within the container, making them visible and functional in your graphical user interface (GUI) applications.
* dispose: a method used in Java, particularly in GUI applications, to release and free up system resources associated with a window or graphical component. It's often used with objects like JFrame or JDialog to close and clean up resources when a window is no longer needed, typically when the user closes the window. This helps to free memory and other resources used by the window and is an essential part of managing resources efficiently in graphical applications.
* setForeground: a Java method for changing the text or graphics color of a GUI component.
* setVisible: a method in Java used to control the visibility of a graphical component, such as a window (e.g., JFrame) or other GUI elements. When you call setVisible(true) on a component, it becomes visible on the screen, allowing the user to see and interact with it. Conversely, calling setVisible(false) hides the component, removing it from view. This method is often used to show or hide GUI components based on user actions or application logic.
* getText: a method in Java used with components like text fields (e.g., JTextField) and text areas to retrieve the text content entered or displayed within the component. It allows you to access the text data for further processing or manipulation within your Java program.
* setText: a method in Java used with components like text fields, labels, and text areas to set the text content that should be displayed within the component. You can use this method to change or update the text displayed to the user. For example, with a JTextField, setText allows you to programmatically change the text in the input field
* isEmpty(): a method in Java used to check whether a data structure or collection is empty, meaning it contains no elements or has a size of zero. This method is commonly used with data structures like strings, arrays, lists, and collections to determine if there is no data or elements present. It returns a boolean value, typically true if the structure is empty and false if it contains one or more elements.
* getSelectedItem(): a Java method used with ComboBoxes to retrieve the currently selected item.
* setStringPainted: The call to setStringPainted causes the progress bar to display, within its bounds, a textual indication of the percentage of the task that has completed
* Timer: a class that is part of the java.util package. It is used to schedule and execute tasks at specified intervals. You can create a Timer object, schedule tasks to run at fixed rates or delays, and manage these tasks efficiently. It is commonly used for automating tasks in the background, such as periodically updating data or triggering events at regular intervals.
* getSource(): a Java method used to get the source object that triggered an event in event handling.
* start(): a Java method that begins the execution of a thread, enabling concurrent processing.
* stop(): Java method used to stop the execution of a thread.
* setDefaultCloseOperation: a Java method for specifying the behavior of a JFrame when the user clicks the close button.
* setRowHeight: a method in Java used with components like JTables to set the height of individual rows in the table. It allows you to control the height of rows in a table, which can be useful for customizing the appearance and layout of tabular data in a graphical user interface (GUI).
* getTableHeader(): a method used in Java with JTables. It allows you to retrieve the table header component associated with a JTable. The table header typically displays the column headers for the table, making it easier for users to identify and manage the columns in the table. You can use getTableHeader() to access and manipulate the properties and behavior of the table header.
* setFocusable(): a method in Java used with components like buttons, text fields, or other graphical elements. It allows you to specify whether a component can receive the input focus or not. When a component is "focusable," it means it can be selected and interacted with using keyboard input. By calling setFocusable(true), you enable the component to receive focus and respond to keyboard events. Conversely, calling setFocusable(false) disables the component from receiving focus and keyboard input.

**Source Code:**

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.JTableHeader;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

class EmployeeManagementSystem {

EmployeeManagementSystem() {

JFrame home = new JFrame("Employee Management System");

home.setSize(1920, 1080);

home.setLayout(null);

JLabel title = new JLabel("Welcome to Employee Management System");

JButton add\_emp = new JButton("Add Employee Details");

JButton view\_emp = new JButton("View Employee Details");

JButton edit\_emp = new JButton("Edit Employee Details");

JButton search\_emp = new JButton("Search Employee");

JButton exit = new JButton("Exit");

title.setBounds(425, 50, 700, 100);

add\_emp.setBounds(600, 150, 300, 75);

view\_emp.setBounds(600, 250, 300, 75);

edit\_emp.setBounds(600, 350, 300, 75);

search\_emp.setBounds(600, 450, 300, 75);

exit.setBounds(600, 550, 300, 75);

add\_emp.setFocusPainted(false);

view\_emp.setFocusPainted(false);

edit\_emp.setFocusPainted(false);

search\_emp.setFocusPainted(false);

exit.setFocusPainted(false);

title.setFont(new Font("Times New Roman", Font.BOLD, 35));

add\_emp.setFont(new Font("Times New Roman", Font.PLAIN, 22));

view\_emp.setFont(new Font("Times New Roman", Font.PLAIN, 22));

edit\_emp.setFont(new Font("Times New Roman", Font.PLAIN, 22));

search\_emp.setFont(new Font("Times New Roman", Font.PLAIN, 22));

exit.setFont(new Font("Times New Roman", Font.PLAIN, 22));

exit.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

home.add(title);

home.add(add\_emp);

home.add(view\_emp);

home.add(edit\_emp);

home.add(search\_emp);

home.add(exit);

add\_emp.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

home.dispose();

JFrame emp\_add = new JFrame("Employee Management System - Add Employee");

emp\_add.setSize(1920, 1080);

emp\_add.setLayout(null);

JLabel id = new JLabel("Enter Emp ID:");

JTextField id\_ip = new JTextField();

JLabel id\_er = new JLabel();

JLabel name = new JLabel("Enter Name:");

JTextField name\_ip = new JTextField();

JLabel name\_er = new JLabel();

JLabel phone = new JLabel("Enter Phone Number:");

JTextField phone\_ip = new JTextField();

JLabel phone\_er = new JLabel();

JLabel email = new JLabel("Enter Email Address:");

JTextField email\_ip = new JTextField(30);

JLabel email\_er = new JLabel();

JLabel age = new JLabel("Enter Age:");

JTextField age\_ip = new JTextField();

JLabel age\_er = new JLabel();

JLabel gender = new JLabel("Select Gender:");

JComboBox genderval = new JComboBox();

genderval.addItem("- Select -");

genderval.addItem("Male");

genderval.addItem("Female");

genderval.setFocusable(false);

JLabel gender\_er = new JLabel();

JLabel desi = new JLabel("Enter Designation:");

JTextField desi\_ip = new JTextField();

JLabel desi\_er = new JLabel();

JLabel sal = new JLabel("Enter Salary (Rs):");

JTextField sal\_ip = new JTextField();

JLabel sal\_er = new JLabel();

JButton back = new JButton("Back");

JButton submit = new JButton("Submit");

id.setFont(new Font("Times New Roman", Font.PLAIN, 25));

id\_ip.setFont(new Font("Times New Roman", Font.PLAIN, 25));

id\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

name.setFont(new Font("Times New Roman", Font.PLAIN, 25));

name\_ip.setFont(new Font("Times New Roman", Font.PLAIN, 25));

name\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

phone.setFont(new Font("Times New Roman", Font.PLAIN, 25));

phone\_ip.setFont(new Font("Times New Roman", Font.PLAIN, 25));

phone\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

email.setFont(new Font("Times New Roman", Font.PLAIN, 25));

email\_ip.setFont(new Font("Times New Roman", Font.PLAIN, 25));

email\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

age.setFont(new Font("Times New Roman", Font.PLAIN, 25));

age\_ip.setFont(new Font("Times New Roman", Font.PLAIN, 25));

age\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

gender.setFont(new Font("Times New Roman", Font.PLAIN, 25));

genderval.setFont(new Font("Times New Roman", Font.PLAIN, 25));

gender\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

desi.setFont(new Font("Times New Roman", Font.PLAIN, 25));

desi\_ip.setFont(new Font("Times New Roman", Font.PLAIN, 25));

desi\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

sal.setFont(new Font("Times New Roman", Font.PLAIN, 25));

sal\_ip.setFont(new Font("Times New Roman", Font.PLAIN, 25));

sal\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

back.setFont(new Font("Times New Roman", Font.PLAIN, 20));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 20));

id.setBounds(100, 50, 260, 50);

id\_ip.setBounds(350, 50, 260, 50);

id\_er.setBounds(350, 95, 260, 30);

name.setBounds(100, 125, 260, 50);

name\_ip.setBounds(350, 125, 260, 50);

name\_er.setBounds(350, 165, 260, 30);

phone.setBounds(100, 200, 260, 50);

phone\_ip.setBounds(350, 200, 260, 50);

phone\_er.setBounds(350, 240, 260, 30);

email.setBounds(100, 275, 260, 50);

email\_ip.setBounds(350, 275, 260, 50);

email\_er.setBounds(350, 315, 260, 30);

age.setBounds(100, 350, 260, 50);

age\_ip.setBounds(350, 350, 260, 50);

age\_er.setBounds(350, 390, 260, 30);

gender.setBounds(100, 425, 260, 50);

genderval.setBounds(350, 425, 260, 50);

gender\_er.setBounds(350, 465, 260, 30);

desi.setBounds(100, 500, 260, 50);

desi\_ip.setBounds(350, 500, 260, 50);

desi\_er.setBounds(350, 540, 260, 30);

sal.setBounds(100, 575, 260, 50);

sal\_ip.setBounds(350, 575, 260, 50);

sal\_er.setBounds(350, 615, 260, 30);

back.setBounds(100, 700, 250, 50);

submit.setBounds(400, 700, 250, 50);

id\_er.setForeground(Color.red);

name\_er.setForeground(Color.red);

phone\_er.setForeground(Color.red);

email\_er.setForeground(Color.red);

age\_er.setForeground(Color.red);

gender\_er.setForeground(Color.red);

desi\_er.setForeground(Color.red);

sal\_er.setForeground(Color.red);

back.setFocusPainted(false);

submit.setFocusPainted(false);

emp\_add.add(id);

emp\_add.add(id\_ip);

emp\_add.add(id\_er);

emp\_add.add(name);

emp\_add.add(name\_ip);

emp\_add.add(name\_er);

emp\_add.add(phone);

emp\_add.add(phone\_ip);

emp\_add.add(phone\_er);

emp\_add.add(email);

emp\_add.add(email\_ip);

emp\_add.add(email\_er);

emp\_add.add(age);

emp\_add.add(age\_ip);

emp\_add.add(age\_er);

emp\_add.add(gender);

emp\_add.add(genderval);

emp\_add.add(gender\_er);

emp\_add.add(desi);

emp\_add.add(desi\_ip);

emp\_add.add(desi\_er);

emp\_add.add(sal);

emp\_add.add(sal\_ip);

emp\_add.add(sal\_er);

emp\_add.add(back);

emp\_add.add(submit);

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

emp\_add.dispose();

home.setVisible(true);

}

});

submit.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

boolean name\_bool = false, phone\_bool = false, email\_bool = false, age\_bool = false,

gender\_bool = false, desi\_bool = false, sal\_bool = false;

// int id\_val = Integer.parseInt(id\_ip.getText());

String name\_val = name\_ip.getText();

String phone\_val = phone\_ip.getText();

String email\_val = email\_ip.getText();

int age\_val = Integer.parseInt(age\_ip.getText());

String gender\_val = (String) genderval.getSelectedItem();

String desi\_val = desi\_ip.getText();

String sal\_val = sal\_ip.getText();

long sal\_long = Long.parseLong(sal\_ip.getText());

for (int i = 0; i < name\_val.length(); i++) {

char ch = name\_val.charAt(i);

if (ch < 'a' && ch > 'z' || ch >= 'A' && ch <= 'Z') {

name\_er.setText("Enter Name in Alphabet");

name\_bool = false;

} else if (!(ch < '0' || ch > '9')) {

name\_er.setText("Enter Name in Alphabet");

name\_bool = false;

} else {

name\_er.setText("");

name\_bool = true;

}

}

for (int i = 0; i < phone\_val.length(); i++) {

char ch = phone\_val.charAt(i);

if (phone\_val.length() != 10) {

phone\_er.setText("Enter a valid phone number (10 Digits)");

phone\_bool = false;

} else if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z') {

phone\_er.setText("Enter a valid phone number (10 Digits)");

phone\_bool = false;

} else {

phone\_er.setText("");

phone\_bool = true;

}

}

int atIndex, dotIndex;

atIndex = email\_val.indexOf('@');

dotIndex = email\_val.indexOf('.');

if (atIndex <= 0 || dotIndex <= atIndex || dotIndex == email\_val.length() - 1) {

email\_er.setText("Email should contain '@' and '.' ");

email\_bool = false;

} else {

email\_er.setText("");

email\_bool = true;

}

if (age\_val < 18) {

age\_er.setText("Enter a valid age (above 18)");

age\_bool = false;

} else {

age\_er.setText("");

age\_bool = true;

}

if ("- Select -".equals(gender\_val)) {

gender\_er.setText("Please Select Your Gender");

gender\_bool = false;

} else {

gender\_er.setText("");

gender\_bool = true;

}

if (desi\_val.isEmpty()) {

desi\_er.setText("Enter Designation");

desi\_bool = false;

} else {

for (int i = 0; i < desi\_val.length(); i++) {

char ch = desi\_val.charAt(i);

if (!(ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z')) {

desi\_er.setText("Enter Designation in Alphabet");

desi\_bool = false;

} else {

desi\_er.setText("");

desi\_bool = true;

}

}

}

for (int i = 0; i < sal\_val.length(); i++) {

char ch = sal\_val.charAt(i);

if (ch < '0' || ch > '9') {

sal\_er.setText("Enter Salary in Numbers");

sal\_bool = false;

} else if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z') {

sal\_er.setText("Enter Salary in Numbers");

sal\_bool = false;

} else {

sal\_er.setText("");

sal\_bool = true;

}

}

if (sal\_long < 0) {

sal\_er.setText("Salary cannot be a negative number");

sal\_bool = false;

} else {

sal\_er.setText("");

sal\_bool = true;

}

if (name\_bool && phone\_bool && email\_bool && age\_bool && desi\_bool && gender\_bool

&& sal\_bool == true) {

JFrame pb = new JFrame("Saving Employee Data.....");

pb.setLayout(null);

pb.setBounds(600, 350, 330, 125);

JProgressBar save = new JProgressBar(0, 100);

pb.add(save);

save.setStringPainted(true);

save.setBounds(50, 30, 220, 30);

Timer timer = new Timer(500, new ActionListener() {

int progress = 0;

public void actionPerformed(ActionEvent e) {

if (progress <= 100) {

save.setValue(progress);

save.setString(progress + "%");

progress += 20; // Simulate progress by incrementing 20% at a time

} else {

((Timer) e.getSource()).stop();

pb.dispose();

JFrame saved = new JFrame("Success");

saved.setBounds(600, 350, 250, 150);

saved.setLayout(null);

JLabel success\_msg = new JLabel("Data Saved Successfully!");

success\_msg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

success\_msg.setBounds(30, 25, 200, 30);

JButton ok = new JButton("OK");

ok.setFocusPainted(false);

ok.setBounds(80, 60, 70, 30);

ok.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

saved.dispose();

emp\_add.dispose();

home.setVisible(true);

}

});

saved.add(success\_msg);

saved.add(ok);

saved.setVisible(true);

}

}

});

timer.start(); // Start the timer to update the progress

pb.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

pb.setVisible(true);

}

}

});

emp\_add.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

emp\_add.setVisible(true);

}

});

view\_emp.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JFrame pb = new JFrame("Fetching Employee Data.....");

pb.setLayout(null);

pb.setBounds(600, 350, 330, 125);

JProgressBar save = new JProgressBar(0, 100);

pb.add(save);

save.setStringPainted(true);

save.setBounds(50, 30, 220, 30);

Timer timer = new Timer(200, new ActionListener() {

int progress = 0;

public void actionPerformed(ActionEvent e) {

if (progress <= 100) {

save.setValue(progress);

save.setString(progress + "%");

progress += 25; // Simulate progress by incrementing 20% at a time

} else {

((Timer) e.getSource()).stop();

pb.dispose();

home.dispose();

JFrame vmain = new JFrame("Employee Management System - View Employee Details");

vmain.setSize(1920, 1080);

vmain.setLayout(null);

// Create column names and data

String[] columnNames = { "Emp ID", "Name", "Phone Number", "Email Address", "Age", "Gender",

"Designation", "Salary" };

Object[][] data = {

{ 1, "Abc", "1234567890", "abc@gmail.com", "18", "Male", "Manager", 1000000 },

{ 2, "Dhruv", "8055128556", "adruv@gmail.com", "18", "Male", "CEO", 1500000 },

{ 3, "Samarthya", "7249336212", "samu@gmail.com", "19", "Male", "Developer", 800000 },

{ 4, "Atharva", "9129215879", "atharva@gmail.com", "18", "Male", "Tester", 800000 },

{ 5, "Mohit", "7824692159", "mohit@gmail.com", "18", "Male", "Designer", 750000 }

};

DefaultTableModel model = new DefaultTableModel(data, columnNames);

JTable emp\_data = new JTable(model);

// Create a JScrollPane and add the JTable to it

JScrollPane scrollPane = new JScrollPane(emp\_data);

scrollPane.setBounds(30, 100, 1300, 500);

// Set the font for column headers

JTableHeader header = emp\_data.getTableHeader();

header.setFont(new Font("Times New Roman", Font.BOLD, 20));

// Set the font for table cells

emp\_data.setFont(new Font("Times New Roman", Font.PLAIN, 20));

emp\_data.setRowHeight(20);

// Create a JButton for navigation

JButton backbtn = new JButton("Back");

backbtn.setBounds(150, 600, 150, 75);

backbtn.setFont(new Font("Times New Roman", Font.PLAIN, 25));

backbtn.setFocusPainted(false);

vmain.add(scrollPane);

vmain.add(backbtn);

backbtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

vmain.dispose();

home.setVisible(true);

}

});

vmain.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

vmain.setVisible(true);

}

}

});

timer.start(); // Start the timer to update the progress

pb.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

pb.setVisible(true);

}

});

edit\_emp.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

home.dispose();

JFrame emain = new JFrame("Employee Management System - Search Employee");

emain.setSize(1920, 1080);

emain.setLayout(null);

JLabel emp\_id = new JLabel("Enter Employee ID:");

JTextField emp\_id\_ip = new JTextField();

JButton cancel = new JButton("Back");

JButton submit = new JButton("Submit");

emp\_id.setFont(new Font("Times New Roman", Font.PLAIN, 35));

emp\_id\_ip.setFont(new Font("Times New Roman", Font.PLAIN, 35));

cancel.setFont(new Font("Times New Roman", Font.PLAIN, 25));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 25));

emp\_id.setBounds(400, 200, 300, 75);

emp\_id\_ip.setBounds(700, 222, 300, 40);

cancel.setBounds(400, 320, 200, 50);

submit.setBounds(800, 320, 200, 50);

cancel.setFocusPainted(false);

submit.setFocusPainted(false);

emain.add(emp\_id);

emain.add(emp\_id\_ip);

emain.add(cancel);

emain.add(submit);

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int id\_val = Integer.parseInt(emp\_id\_ip.getText());

if (id\_val != 10) {

emain.dispose();

JFrame edit\_emp = new JFrame("Employee Management System - Edit Employee Details");

edit\_emp.setLayout(null);

edit\_emp.setSize(1920, 1080);

JLabel edit\_msg = new JLabel("What you want to edit?");

edit\_msg.setBounds(300, 200, 550, 75);

edit\_msg.setFont(new Font("Times New Roman", Font.PLAIN, 35));

JComboBox choice = new JComboBox();

choice.setBounds(650, 215, 300, 50);

choice.setFont(new Font("Times New Roman", Font.PLAIN, 35));

choice.addItem("Select Field");

choice.addItem("Name");

choice.addItem("Phone Number");

choice.addItem("Email Address");

choice.addItem("Age");

choice.addItem("Gender");

choice.addItem("Designation");

choice.addItem("Salary");

choice.setFocusable(false);

JButton back = new JButton("Back");

back.setFont(new Font("Times New Roman", Font.PLAIN, 25));

back.setBounds(300, 350, 200, 50);

back.setFocusPainted(false);

JButton next = new JButton("Next");

next.setFocusPainted(false);

next.setBounds(750, 350, 200, 50);

next.setFont(new Font("Times New Roman", Font.PLAIN, 25));

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

edit\_emp.dispose();

emain.setVisible(true);

}

});

next.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String selected = (String) choice.getSelectedItem();

if (selected == "Name") {

JFrame popup = new JFrame("Edit Name");

JLabel ename = new JLabel("Name:");

ename.setFont(new Font("Times New Roman", Font.PLAIN, 20));

ename.setBounds(10, 20, 260, 50);

JTextField name\_tf = new JTextField();

name\_tf.setFont(new Font("Times New Roman", Font.PLAIN, 20));

name\_tf.setBounds(100, 30, 260, 30);

JLabel name\_er = new JLabel();

name\_er.setBounds(100, 60, 300, 30);

name\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

name\_er.setForeground(Color.red);

JButton back = new JButton("Back");

JButton submit = new JButton("Submit");

back.setFocusPainted(false);

submit.setFocusPainted(false);

back.setBounds(25, 150, 150, 30);

submit.setBounds(230, 150, 150, 30);

back.setFont(new Font("Times New Roman", Font.PLAIN, 20));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 20));

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

popup.dispose();

edit\_emp.setVisible(true);

}

});

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

boolean name\_bool = false;

String name\_val = name\_tf.getText();

for (int i = 0; i < name\_val.length(); i++) {

char ch = name\_val.charAt(i);

if (ch < 'a' && ch > 'z' || ch >= 'A' && ch <= 'Z') {

name\_er.setText("Enter Name in Alphabet");

name\_bool = false;

} else if (!(ch < '0' || ch > '9')) {

name\_er.setText("Enter Name in Alphabet");

name\_bool = false;

} else {

name\_er.setText("");

name\_bool = true;

}

}

if (name\_bool == true) {

JFrame dialog = new JFrame("Success");

JLabel popupMsg = new JLabel("Employee Deatails Updated!");

popupMsg.setBounds(20, 10, 300, 50);

popupMsg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

dialog.add(popupMsg);

JButton button = new JButton("OK");

button.setBounds(120, 60, 70, 20);

button.setFont(new Font("Times New Roman", Font.PLAIN, 20));

button.setFocusPainted(false);

button.addActionListener(actionEvent2 -> {

dialog.dispose();

popup.dispose();

edit\_emp.dispose();

home.setVisible(true);

});

dialog.add(button);

dialog.setLayout(null);

dialog.setBounds(500, 300, 350, 150);

dialog.setVisible(true);

}

}

});

popup.setLayout(null);

popup.setBounds(455, 250, 425, 300);

popup.add(ename);

popup.add(name\_tf);

popup.add(name\_er);

popup.add(back);

popup.add(submit);

popup.setVisible(true);

} else if (selected == "Phone Number") {

JFrame popup = new JFrame("Edit Phone Number");

JLabel ephone = new JLabel("Phone Number:");

ephone.setFont(new Font("Times New Roman", Font.PLAIN, 20));

ephone.setBounds(10, 20, 260, 50);

JTextField phone\_tf = new JTextField();

phone\_tf.setFont(new Font("Times New Roman", Font.PLAIN, 20));

phone\_tf.setBounds(150, 30, 230, 30);

JLabel phone\_er = new JLabel();

phone\_er.setBounds(150, 60, 300, 30);

phone\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

phone\_er.setForeground(Color.red);

JButton back = new JButton("Back");

JButton submit = new JButton("Submit");

back.setFocusPainted(false);

submit.setFocusPainted(false);

back.setBounds(25, 150, 150, 30);

submit.setBounds(230, 150, 150, 30);

back.setFont(new Font("Times New Roman", Font.PLAIN, 20));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 20));

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

popup.dispose();

edit\_emp.setVisible(true);

}

});

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

boolean phone\_bool = false;

String phone\_val = phone\_tf.getText();

for (int i = 0; i < phone\_val.length(); i++) {

char ch = phone\_val.charAt(i);

if (phone\_val.length() != 10) {

phone\_er.setText("Enter a valid phone number (10 Digits)");

phone\_bool = false;

} else if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z') {

phone\_er.setText("Enter a valid phone number (10 Digits)");

phone\_bool = false;

} else {

phone\_er.setText("");

phone\_bool = true;

}

}

if (phone\_bool == true) {

JFrame dialog = new JFrame("Success");

JLabel popupMsg = new JLabel("Employee Deatails Updated!");

popupMsg.setBounds(20, 10, 300, 50);

popupMsg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

dialog.add(popupMsg);

JButton button = new JButton("OK");

button.setBounds(120, 60, 70, 20);

button.setFont(new Font("Times New Roman", Font.PLAIN, 20));

button.setFocusPainted(false);

button.addActionListener(actionEvent2 -> {

dialog.dispose();

popup.dispose();

edit\_emp.dispose();

home.setVisible(true);

});

dialog.add(button);

dialog.setLayout(null);

dialog.setBounds(500, 300, 350, 150);

dialog.setVisible(true);

}

}

});

popup.setLayout(null);

popup.setBounds(455, 250, 425, 300);

popup.add(ephone);

popup.add(phone\_tf);

popup.add(phone\_er);

popup.add(back);

popup.add(submit);

popup.setVisible(true);

} else if (selected == "Email Address") {

JFrame popup = new JFrame("Edit Email Address");

JLabel eemail = new JLabel("Email:");

eemail.setFont(new Font("Times New Roman", Font.PLAIN, 20));

eemail.setBounds(10, 20, 260, 50);

JTextField email\_tf = new JTextField();

email\_tf.setFont(new Font("Times New Roman", Font.PLAIN, 20));

email\_tf.setBounds(100, 30, 260, 30);

JLabel email\_er = new JLabel();

email\_er.setBounds(100, 60, 300, 30);

email\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

email\_er.setForeground(Color.red);

JButton back = new JButton("Back");

JButton submit = new JButton("Submit");

back.setFocusPainted(false);

submit.setFocusPainted(false);

back.setBounds(25, 150, 150, 30);

submit.setBounds(230, 150, 150, 30);

back.setFont(new Font("Times New Roman", Font.PLAIN, 20));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 20));

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

popup.dispose();

edit\_emp.setVisible(true);

}

});

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

boolean email\_bool = false;

String email\_val = email\_tf.getText();

int atIndex, dotIndex;

atIndex = email\_val.indexOf('@');

dotIndex = email\_val.indexOf('.');

if (atIndex <= 0 || dotIndex <= atIndex

|| dotIndex == email\_val.length() - 1) {

email\_er.setText("Email should contain '@' and '.' ");

email\_bool = false;

} else {

email\_er.setText("");

email\_bool = true;

}

if (email\_bool == true) {

JFrame dialog = new JFrame("Success");

JLabel popupMsg = new JLabel("Employee Deatails Updated!");

popupMsg.setBounds(20, 10, 300, 50);

popupMsg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

dialog.add(popupMsg);

JButton button = new JButton("OK");

button.setBounds(120, 60, 70, 20);

button.setFont(new Font("Times New Roman", Font.PLAIN, 20));

button.setFocusPainted(false);

button.addActionListener(actionEvent2 -> {

dialog.dispose();

popup.dispose();

edit\_emp.dispose();

home.setVisible(true);

});

dialog.add(button);

dialog.setLayout(null);

dialog.setBounds(500, 300, 350, 150);

dialog.setVisible(true);

}

}

});

popup.setLayout(null);

popup.setBounds(455, 250, 425, 300);

popup.add(eemail);

popup.add(email\_tf);

popup.add(email\_er);

popup.add(back);

popup.add(submit);

popup.setVisible(true);

} else if (selected == "Age") {

JFrame popup = new JFrame("Edit Age");

JLabel eage = new JLabel("Age:");

eage.setFont(new Font("Times New Roman", Font.PLAIN, 20));

eage.setBounds(30, 20, 260, 50);

JTextField age\_tf = new JTextField();

age\_tf.setFont(new Font("Times New Roman", Font.PLAIN, 20));

age\_tf.setBounds(100, 30, 260, 30);

JLabel age\_er = new JLabel();

age\_er.setBounds(100, 60, 300, 30);

age\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

age\_er.setForeground(Color.red);

JButton back = new JButton("Back");

JButton submit = new JButton("Submit");

back.setFocusPainted(false);

submit.setFocusPainted(false);

back.setBounds(25, 150, 150, 30);

submit.setBounds(230, 150, 150, 30);

back.setFont(new Font("Times New Roman", Font.PLAIN, 20));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 20));

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

popup.dispose();

edit\_emp.setVisible(true);

}

});

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

boolean age\_bool = false;

int age\_val = Integer.parseInt(age\_tf.getText());

if (age\_val <= 18) {

age\_er.setText("Enter a valid age (above 18)");

age\_bool = false;

} else {

age\_er.setText("");

age\_bool = true;

}

if (age\_bool == true) {

JFrame dialog = new JFrame("Success");

JLabel popupMsg = new JLabel("Employee Deatails Updated!");

popupMsg.setBounds(20, 10, 300, 50);

popupMsg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

dialog.add(popupMsg);

JButton button = new JButton("OK");

button.setBounds(120, 60, 70, 20);

button.setFont(new Font("Times New Roman", Font.PLAIN, 20));

button.setFocusPainted(false);

button.addActionListener(actionEvent2 -> {

dialog.dispose();

popup.dispose();

edit\_emp.dispose();

home.setVisible(true);

});

dialog.add(button);

dialog.setLayout(null);

dialog.setBounds(500, 300, 350, 150);

dialog.setVisible(true);

}

}

});

popup.setLayout(null);

popup.setBounds(455, 250, 425, 300);

popup.add(eage);

popup.add(age\_tf);

popup.add(age\_er);

popup.add(back);

popup.add(submit);

popup.setVisible(true);

} else if (selected == "Gender") {

JFrame popup = new JFrame("Edit Gender");

JLabel gender = new JLabel("Select Gender:");

JComboBox genderval = new JComboBox();

genderval.addItem("- Select -");

genderval.addItem("Male");

genderval.addItem("Female");

genderval.setFocusable(false);

JLabel gender\_er = new JLabel();

gender.setFont(new Font("Times New Roman", Font.PLAIN, 25));

genderval.setFont(new Font("Times New Roman", Font.PLAIN, 25));

gender\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

gender.setBounds(10, 20, 260, 50);

genderval.setBounds(170, 30, 200, 30);

gender\_er.setBounds(170, 60, 300, 30);

gender\_er.setForeground(Color.red);

JButton back = new JButton("Back");

JButton submit = new JButton("Submit");

back.setFocusPainted(false);

submit.setFocusPainted(false);

back.setBounds(25, 150, 150, 30);

submit.setBounds(230, 150, 150, 30);

back.setFont(new Font("Times New Roman", Font.PLAIN, 20));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 20));

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

popup.dispose();

edit\_emp.setVisible(true);

}

});

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

boolean gender\_bool = false;

String gender\_val = (String) genderval.getSelectedItem();

if ("- Select -".equals(gender\_val)) {

gender\_er.setText("Please Select Your Gender");

gender\_bool = false;

} else {

gender\_er.setText("");

gender\_bool = true;

}

if (gender\_bool == true) {

JFrame dialog = new JFrame("Success");

JLabel popupMsg = new JLabel("Employee Deatails Updated!");

popupMsg.setBounds(20, 10, 300, 50);

popupMsg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

dialog.add(popupMsg);

JButton button = new JButton("OK");

button.setBounds(120, 60, 70, 20);

button.setFont(new Font("Times New Roman", Font.PLAIN, 20));

button.setFocusPainted(false);

button.addActionListener(actionEvent2 -> {

dialog.dispose();

popup.dispose();

edit\_emp.dispose();

home.setVisible(true);

});

dialog.add(button);

dialog.setLayout(null);

dialog.setBounds(500, 300, 350, 150);

dialog.setVisible(true);

}

}

});

popup.setLayout(null);

popup.setBounds(455, 250, 425, 300);

popup.add(genderval);

popup.add(gender\_er);

popup.add(gender);

popup.add(back);

popup.add(submit);

popup.setVisible(true);

} else if (selected == "Designation") {

JFrame popup = new JFrame("Edit Designation");

JLabel edesi = new JLabel("Desination:");

edesi.setFont(new Font("Times New Roman", Font.PLAIN, 20));

edesi.setBounds(10, 20, 260, 50);

JTextField desi\_tf = new JTextField();

desi\_tf.setFont(new Font("Times New Roman", Font.PLAIN, 20));

desi\_tf.setBounds(100, 30, 260, 30);

JLabel desi\_er = new JLabel();

desi\_er.setBounds(100, 60, 300, 30);

desi\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

desi\_er.setForeground(Color.red);

JButton back = new JButton("Back");

JButton submit = new JButton("Submit");

back.setFocusPainted(false);

submit.setFocusPainted(false);

back.setBounds(25, 150, 150, 30);

submit.setBounds(230, 150, 150, 30);

back.setFont(new Font("Times New Roman", Font.PLAIN, 20));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 20));

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

popup.dispose();

edit\_emp.setVisible(true);

}

});

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

boolean desi\_bool = false;

String desi\_val = desi\_tf.getText();

if (desi\_val.isEmpty()) {

desi\_er.setText("Enter Designation");

desi\_bool = false;

} else {

for (int i = 0; i < desi\_val.length(); i++) {

char ch = desi\_val.charAt(i);

if (!(ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z')) {

desi\_er.setText("Enter Designation in Alphabet");

desi\_bool = false;

} else {

desi\_er.setText("");

desi\_bool = true;

}

}

}

if (desi\_bool == true) {

JFrame dialog = new JFrame("Success");

JLabel popupMsg = new JLabel("Employee Deatails Updated!");

popupMsg.setBounds(20, 10, 300, 50);

popupMsg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

dialog.add(popupMsg);

JButton button = new JButton("OK");

button.setBounds(120, 60, 70, 20);

button.setFont(new Font("Times New Roman", Font.PLAIN, 20));

button.setFocusPainted(false);

button.addActionListener(actionEvent2 -> {

dialog.dispose();

popup.dispose();

edit\_emp.dispose();

home.setVisible(true);

});

dialog.add(button);

dialog.setLayout(null);

dialog.setBounds(500, 300, 350, 150);

dialog.setVisible(true);

}

}

});

popup.setLayout(null);

popup.setBounds(455, 250, 425, 300);

popup.add(edesi);

popup.add(desi\_tf);

popup.add(desi\_er);

popup.add(back);

popup.add(submit);

popup.setVisible(true);

} else if (selected == "Salary") {

JFrame popup = new JFrame("Edit Salary ");

JLabel esal = new JLabel("Salary:");

esal.setFont(new Font("Times New Roman", Font.PLAIN, 20));

esal.setBounds(10, 20, 260, 50);

JTextField sal\_tf = new JTextField();

sal\_tf.setFont(new Font("Times New Roman", Font.PLAIN, 20));

sal\_tf.setBounds(100, 30, 260, 30);

JLabel sal\_er = new JLabel();

sal\_er.setBounds(100, 60, 300, 30);

sal\_er.setFont(new Font("Times New Roman", Font.PLAIN, 15));

sal\_er.setForeground(Color.red);

JButton back = new JButton("Back");

JButton submit = new JButton("Submit");

back.setFocusPainted(false);

submit.setFocusPainted(false);

back.setBounds(25, 150, 150, 30);

submit.setBounds(230, 150, 150, 30);

back.setFont(new Font("Times New Roman", Font.PLAIN, 20));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 20));

back.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

popup.dispose();

edit\_emp.setVisible(true);

}

});

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

boolean sal\_bool = false;

String sal\_val = sal\_tf.getText();

long sal\_long = Long.parseLong(sal\_tf.getText());

for (int i = 0; i < sal\_val.length(); i++) {

char ch = sal\_val.charAt(i);

if (ch < '0' || ch > '9') {

sal\_er.setText("Enter Salary in Numbers");

sal\_bool = false;

} else if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z') {

sal\_er.setText("Enter Salary in Numbers");

sal\_bool = false;

} else {

sal\_er.setText("");

sal\_bool = true;

}

}

if (sal\_long < 0) {

sal\_er.setText("Salary cannot be a negative number");

sal\_bool = false;

} else {

sal\_er.setText("");

sal\_bool = true;

}

if (sal\_bool == true) {

JFrame dialog = new JFrame("Success");

JLabel popupMsg = new JLabel("Employee Deatails Updated!");

popupMsg.setBounds(20, 10, 300, 50);

popupMsg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

dialog.add(popupMsg);

JButton button = new JButton("OK");

button.setBounds(120, 60, 70, 20);

button.setFont(new Font("Times New Roman", Font.PLAIN, 20));

button.setFocusPainted(false);

button.addActionListener(actionEvent2 -> {

dialog.dispose();

popup.dispose();

edit\_emp.dispose();

home.setVisible(true);

});

dialog.add(button);

dialog.setLayout(null);

dialog.setBounds(500, 300, 350, 150);

dialog.setVisible(true);

}

}

});

popup.setLayout(null);

popup.setBounds(455, 250, 425, 300);

popup.add(esal);

popup.add(sal\_tf);

popup.add(sal\_er);

popup.add(back);

popup.add(submit);

popup.setVisible(true);

} else {

JFrame popup = new JFrame("Invalid Choice");

JLabel popupMsg = new JLabel("Please Select a valid field");

popupMsg.setBounds(20, 10, 300, 50);

popupMsg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

popup.add(popupMsg);

JButton button = new JButton("OK");

button.setBounds(120, 60, 70, 20);

button.setFont(new Font("Times New Roman", Font.PLAIN, 20));

button.setFocusPainted(false);

button.addActionListener(actionEvent2 -> {

popup.dispose();

});

popup.add(button);

popup.setLayout(null);

popup.setBounds(425, 250, 600, 300);

popup.setVisible(true);

}

}

});

edit\_emp.add(edit\_msg);

edit\_emp.add(choice);

edit\_emp.add(back);

edit\_emp.add(next);

edit\_emp.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

edit\_emp.setVisible(true);

}

}

});

cancel.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

emain.dispose();

home.setVisible(true);

}

});

emain.setVisible(true);

}

});

search\_emp.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

home.dispose();

JFrame smain = new JFrame("Employee Management System - Search Employee");

smain.setSize(1920, 1080);

smain.setLayout(null);

JLabel emp\_id = new JLabel("Enter Employee ID:");

JTextField emp\_id\_ip = new JTextField();

JButton cancel = new JButton("Back");

JButton submit = new JButton("Submit");

emp\_id.setFont(new Font("Times New Roman", Font.PLAIN, 35));

emp\_id\_ip.setFont(new Font("Times New Roman", Font.PLAIN, 35));

cancel.setFont(new Font("Times New Roman", Font.PLAIN, 25));

submit.setFont(new Font("Times New Roman", Font.PLAIN, 25));

emp\_id.setBounds(400, 200, 300, 75);

emp\_id\_ip.setBounds(700, 222, 300, 40);

cancel.setBounds(400, 320, 200, 50);

submit.setBounds(800, 320, 200, 50);

cancel.setFocusPainted(false);

submit.setFocusPainted(false);

smain.add(emp\_id);

smain.add(emp\_id\_ip);

smain.add(cancel);

smain.add(submit);

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int id\_val = Integer.parseInt(emp\_id\_ip.getText());

if (id\_val == 10) {

JFrame pb = new JFrame("Fetching Employee Data.....");

pb.setLayout(null);

pb.setBounds(600, 350, 330, 125);

JProgressBar save = new JProgressBar(0, 100);

pb.add(save);

save.setStringPainted(true);

save.setBounds(50, 30, 220, 30);

Timer timer = new Timer(200, new ActionListener() {

int progress = 0;

public void actionPerformed(ActionEvent e) {

if (progress <= 100) {

save.setValue(progress);

save.setString(progress + "%");

progress += 25; // Simulate progress by incrementing 20% at a time

} else {

((Timer) e.getSource()).stop();

pb.dispose();

home.dispose();

JFrame vmain = new JFrame("Employee Management System - View Employee Details");

vmain.setSize(1920, 1080);

vmain.setLayout(null);

String[] columnNames = { "Emp ID", "Name", "Phone Number", "Email Address",

"Age", "Gender", "Designation", "Salary" };

Object[][] data = {

{ 1, "Abc", "1234567890", "abc@gmail.com", "18", "Male", "CEO",

150000 },

};

DefaultTableModel model = new DefaultTableModel(data, columnNames);

JTable emp\_data = new JTable(model);

// Create a JScrollPane and add the JTable to it

JScrollPane scrollPane = new JScrollPane(emp\_data);

scrollPane.setBounds(30, 100, 1300, 500);

// Set the font for column headers

JTableHeader header = emp\_data.getTableHeader();

header.setFont(new Font("Times New Roman", Font.BOLD, 20));

// Set the font for table cells

emp\_data.setFont(new Font("Times New Roman", Font.PLAIN, 20));

emp\_data.setRowHeight(20);

// Create a JButton for navigation

JButton backbtn = new JButton("Back");

backbtn.setBounds(150, 600, 150, 75);

backbtn.setFont(new Font("Times New Roman", Font.PLAIN, 25));

backbtn.setFocusPainted(false);

vmain.add(scrollPane);

vmain.add(backbtn);

backbtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

vmain.dispose();

home.setVisible(true);

}

});

vmain.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

vmain.setVisible(true);

}

}

});

timer.start(); // Start the timer to update the progress

pb.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

pb.setVisible(true);

} else {

JFrame popup = new JFrame("Invalid ID");

JLabel popupMsg = new JLabel("The ID you entered is invalid.");

popupMsg.setBounds(20, 10, 300, 50);

popupMsg.setFont(new Font("Times New Roman", Font.PLAIN, 20));

popup.add(popupMsg);

JButton button = new JButton("OK");

button.setBounds(120, 60, 70, 20);

button.setFont(new Font("Times New Roman", Font.PLAIN, 20));

button.setFocusPainted(false);

button.addActionListener(actionEvent2 -> {

popup.dispose();

});

popup.add(button);

popup.setLayout(null);

popup.setBounds(525, 250, 350, 150);

popup.setVisible(true);

}

}

});

cancel.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

smain.dispose();

home.setVisible(true);

}

});

smain.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

smain.setVisible(true);

}

});

home.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

home.setVisible(true);

}

public static void main(String[] args) {

EmployeeManagementSystem ems = new EmployeeManagementSystem();

}

}

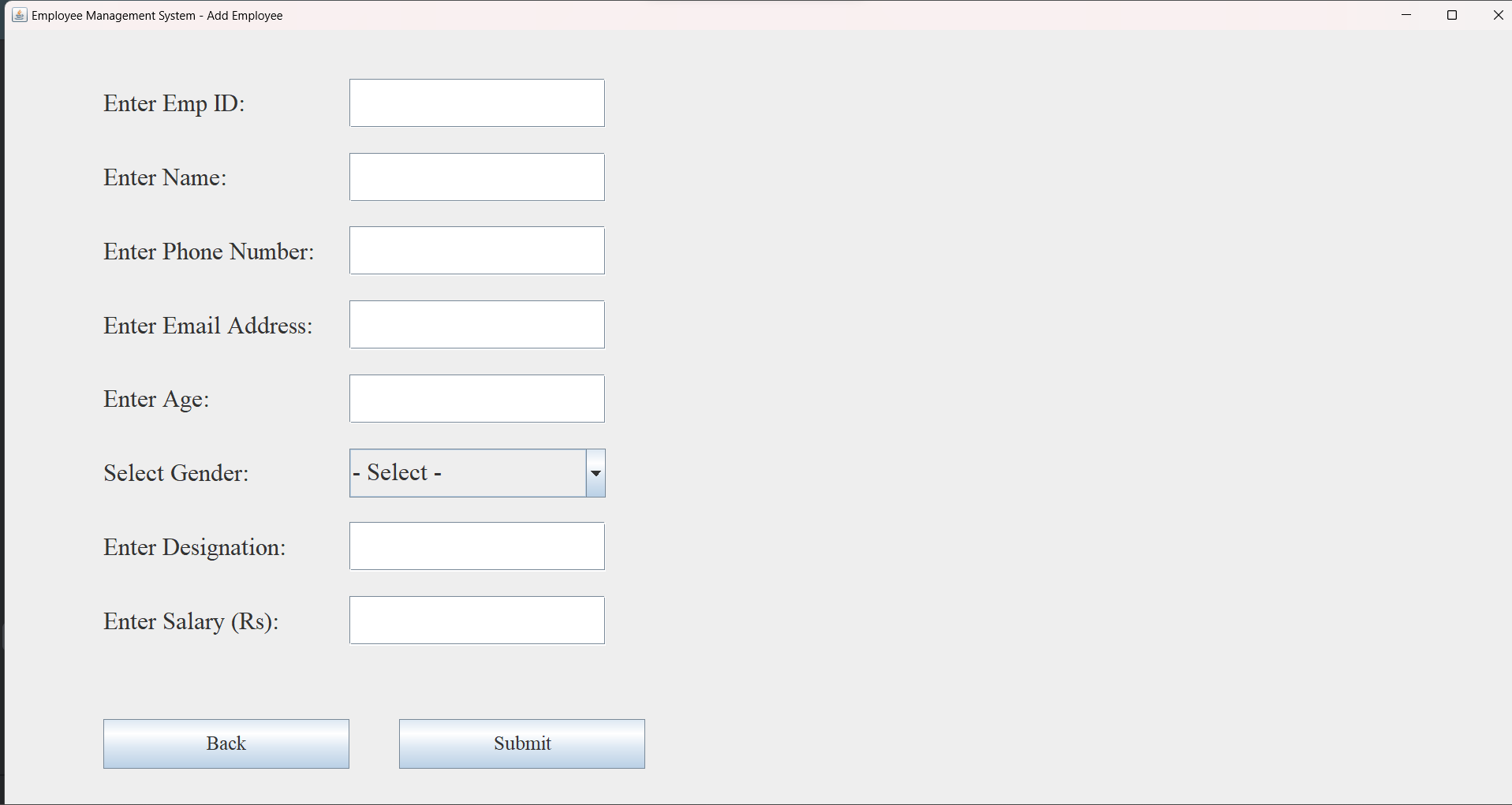
1. **Actual Resources Required:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Name of Resource/Material** | **Specification** | **Quantity** | **Remark** |
| 1 | Computer System | **OS:** Windows 11 (64 bit)  **Processor:** Intel i7 11th generation  **RAM:** 16GB | 1 | - |
| 2 | Software | jdk 11.0 | 1 | - |

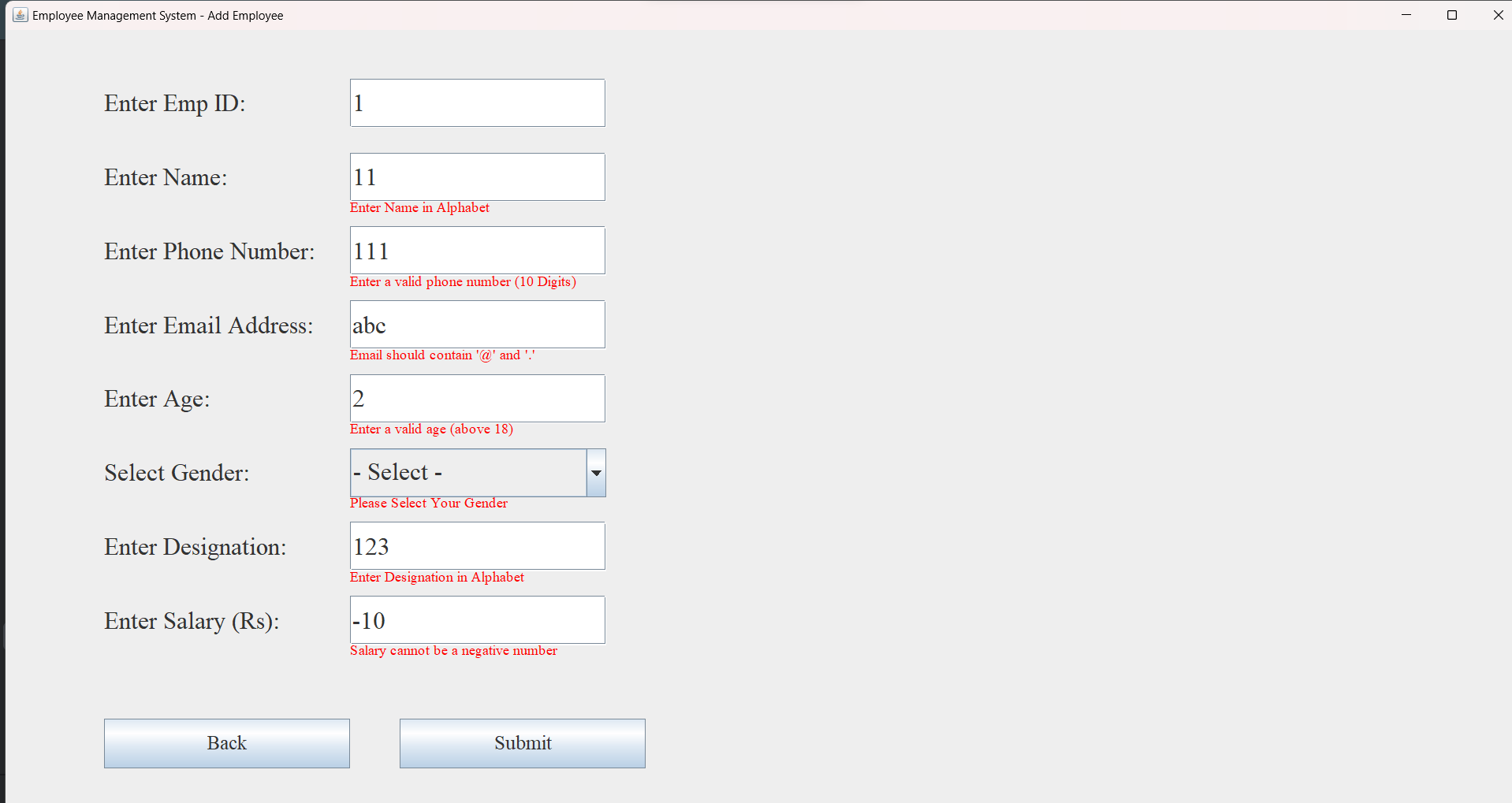
1. **Output of Micro-Project:**
2. **Home Page**

****

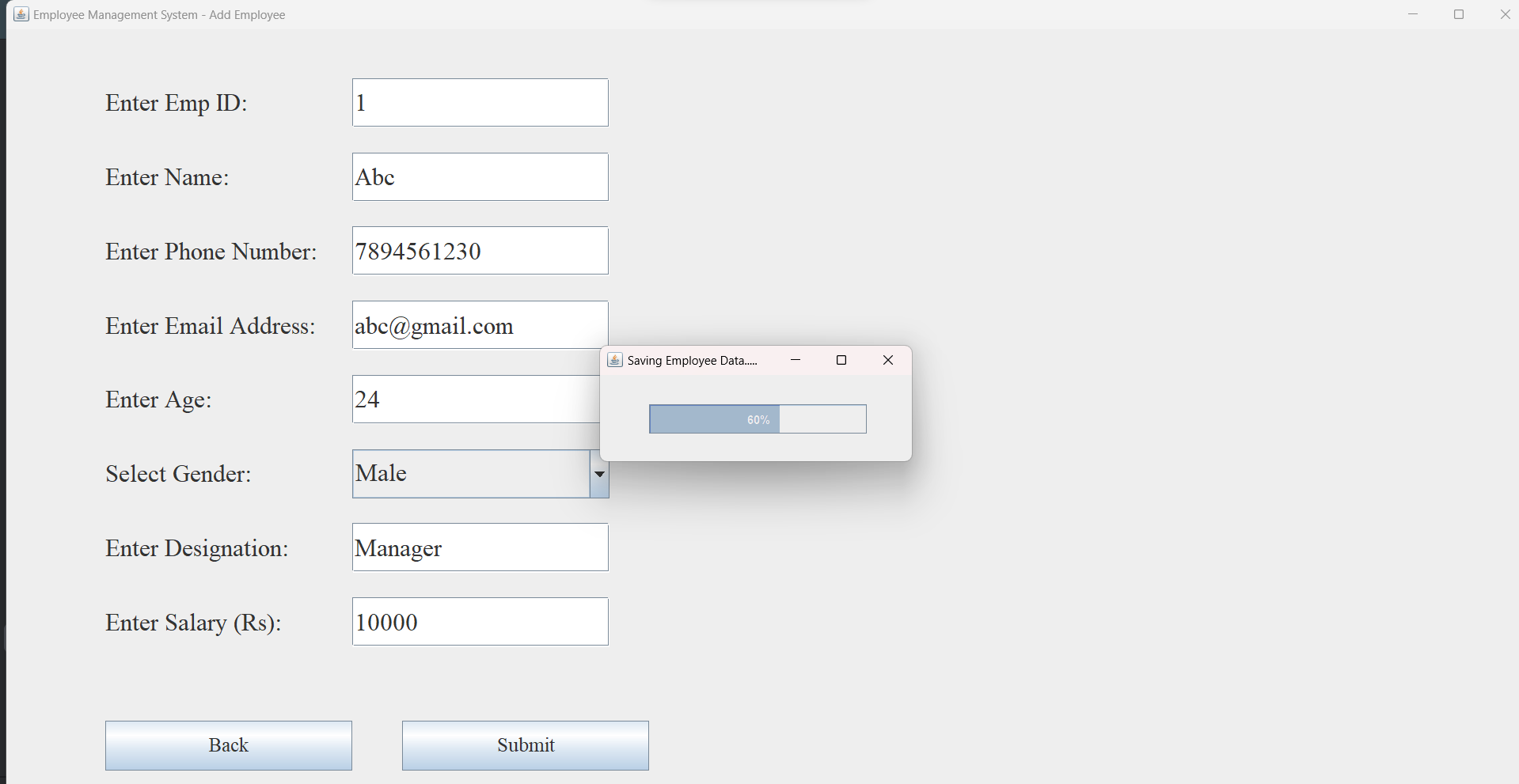
1. **Add Employee Details**

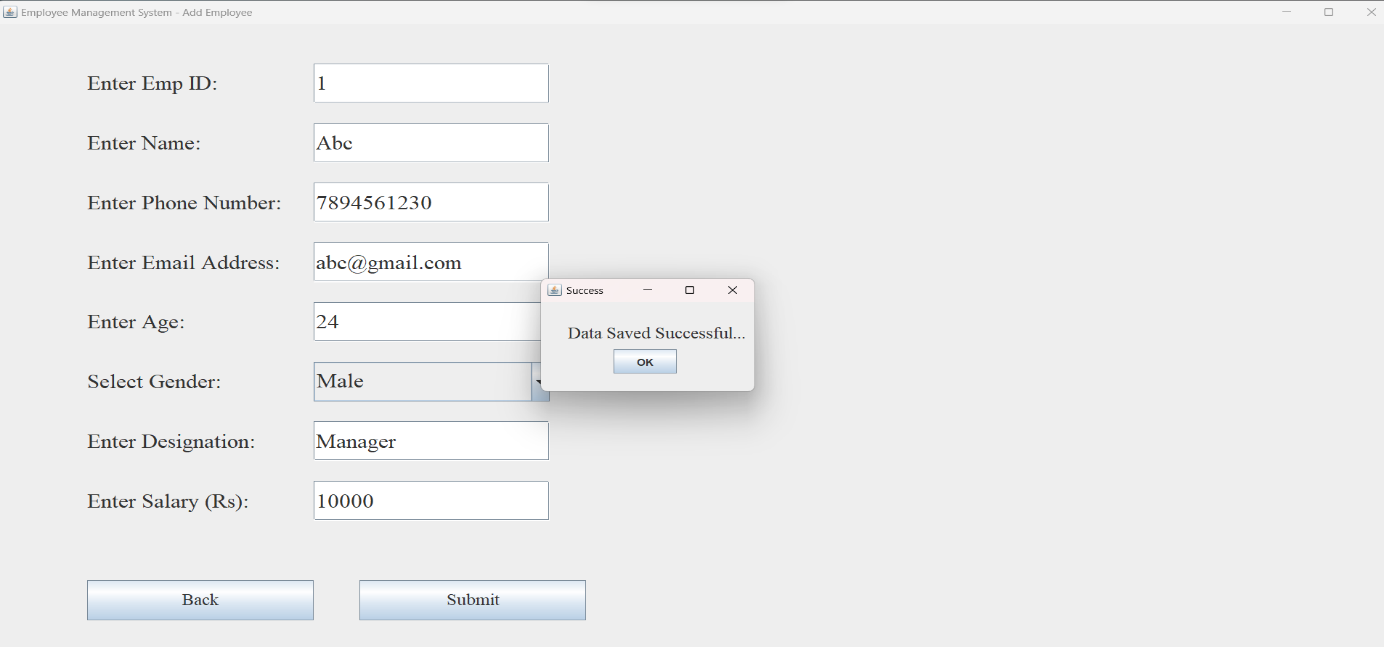
****

1. **Add Employee Details - Validation**

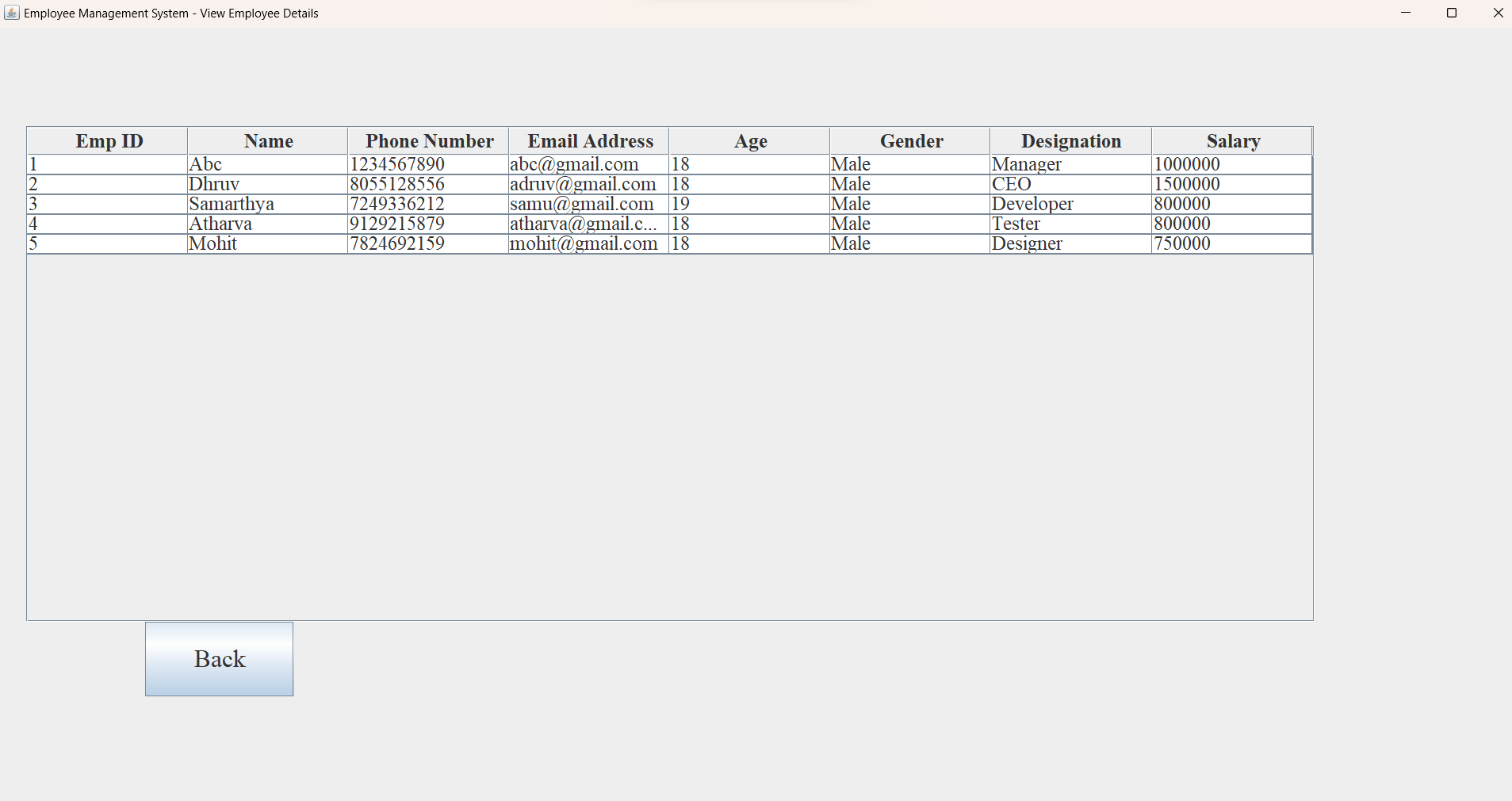
****

1. **Add Employee Details – Loading and Success Message**

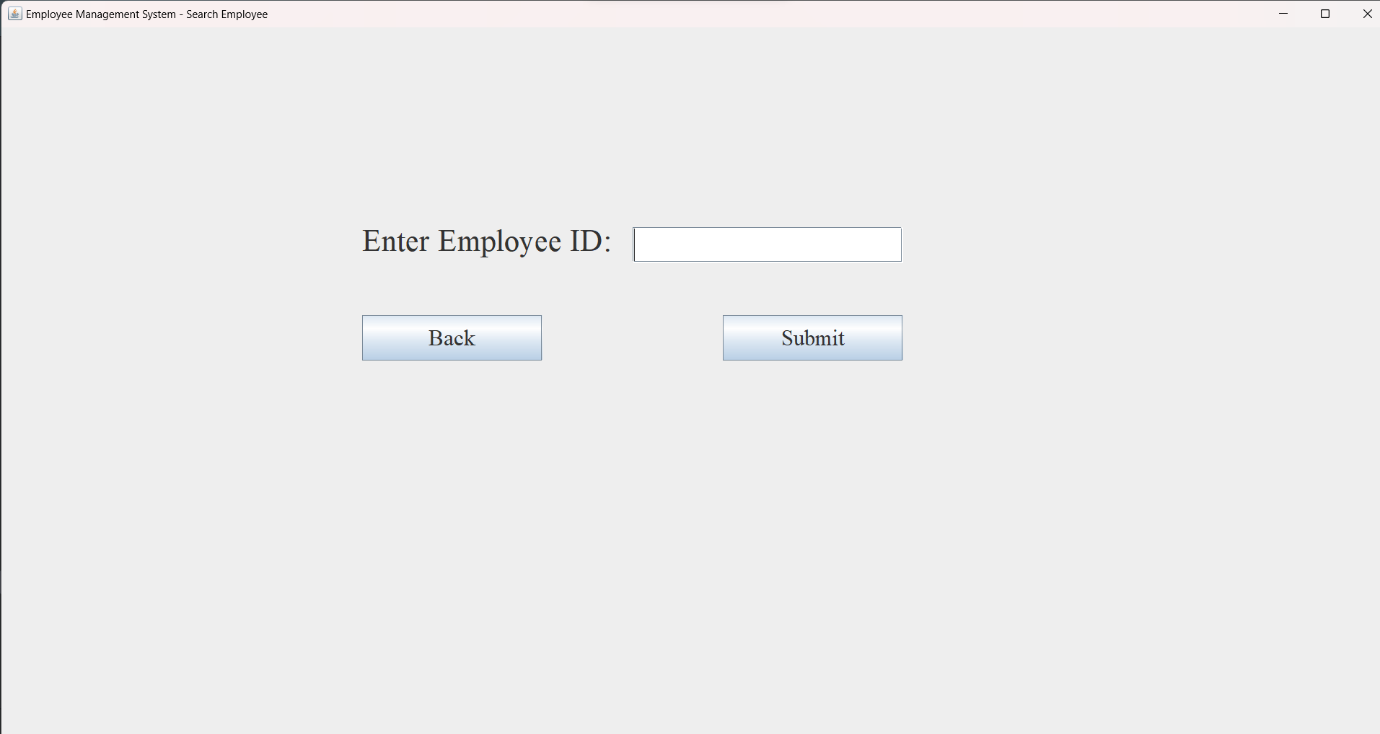
****

****

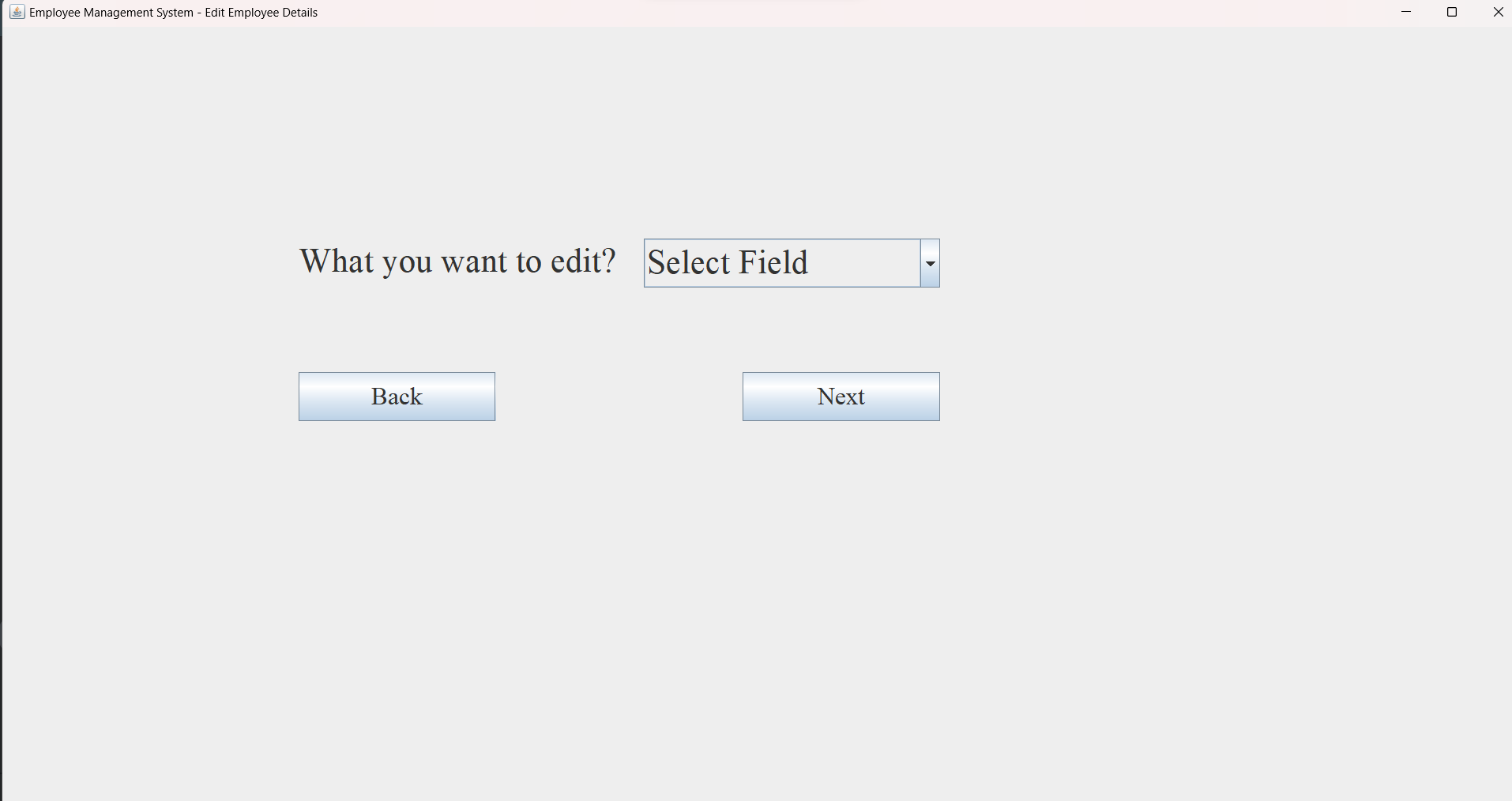
1. **View Employee Details**

****

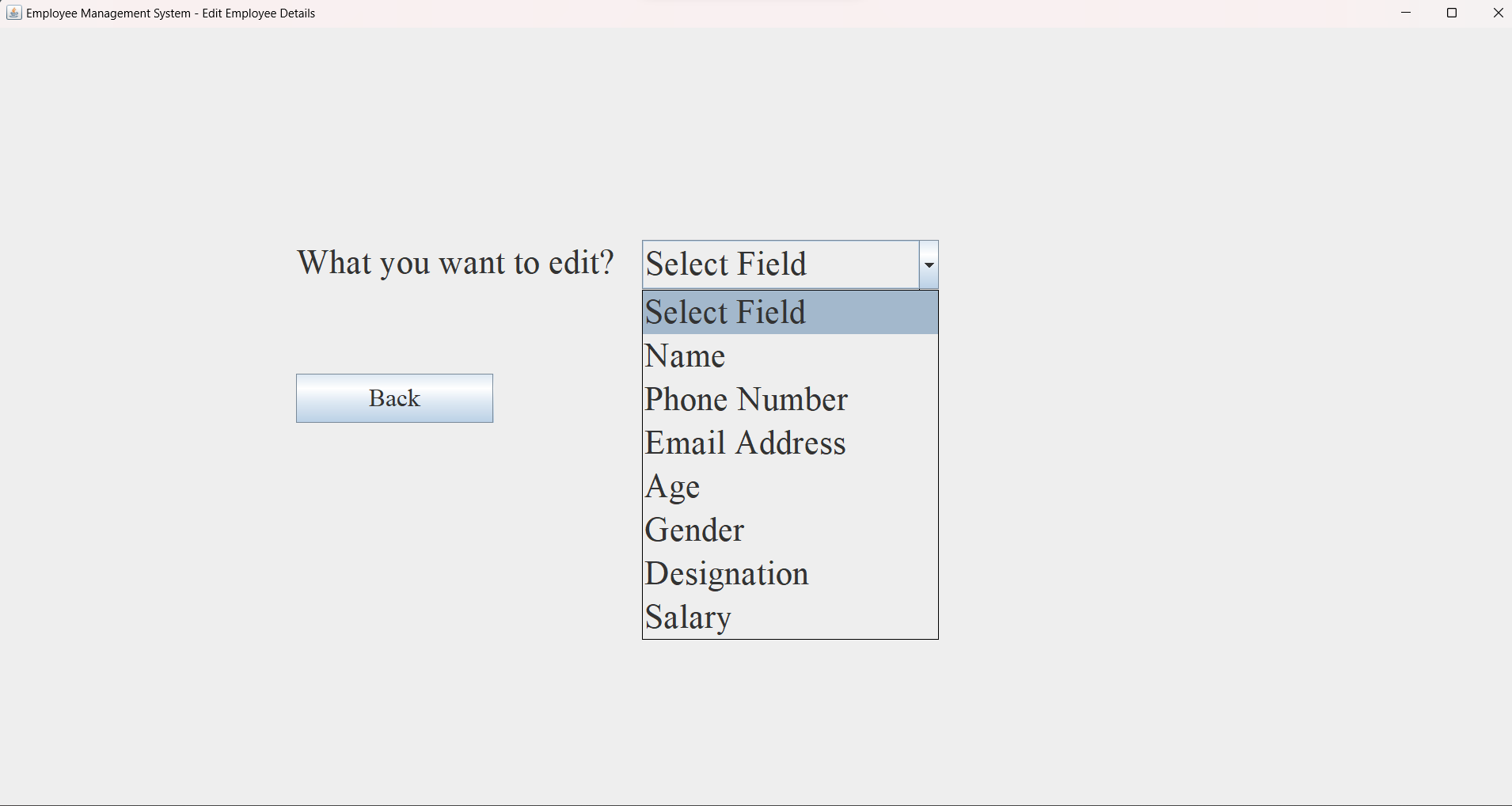
1. **Edit Employee Details**

****

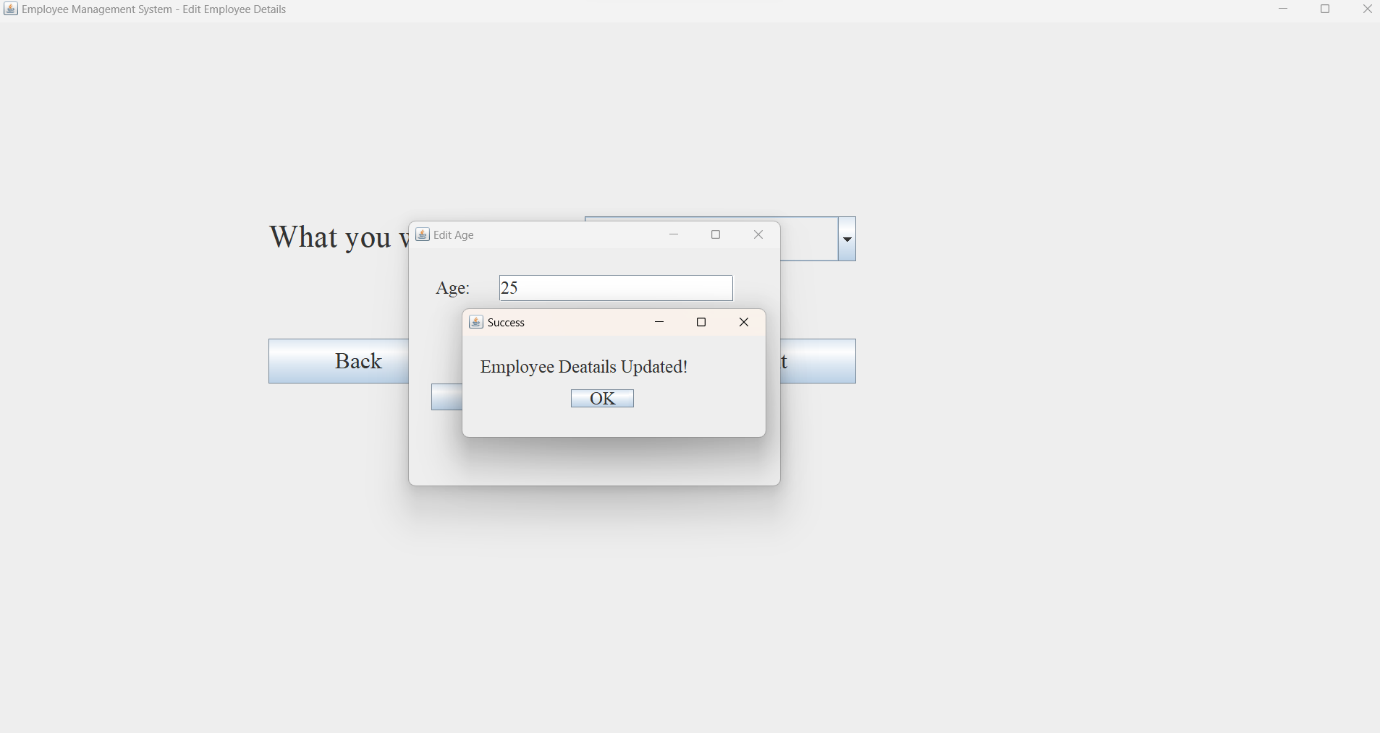
1. **Edit Employee Details - Select Field**

****

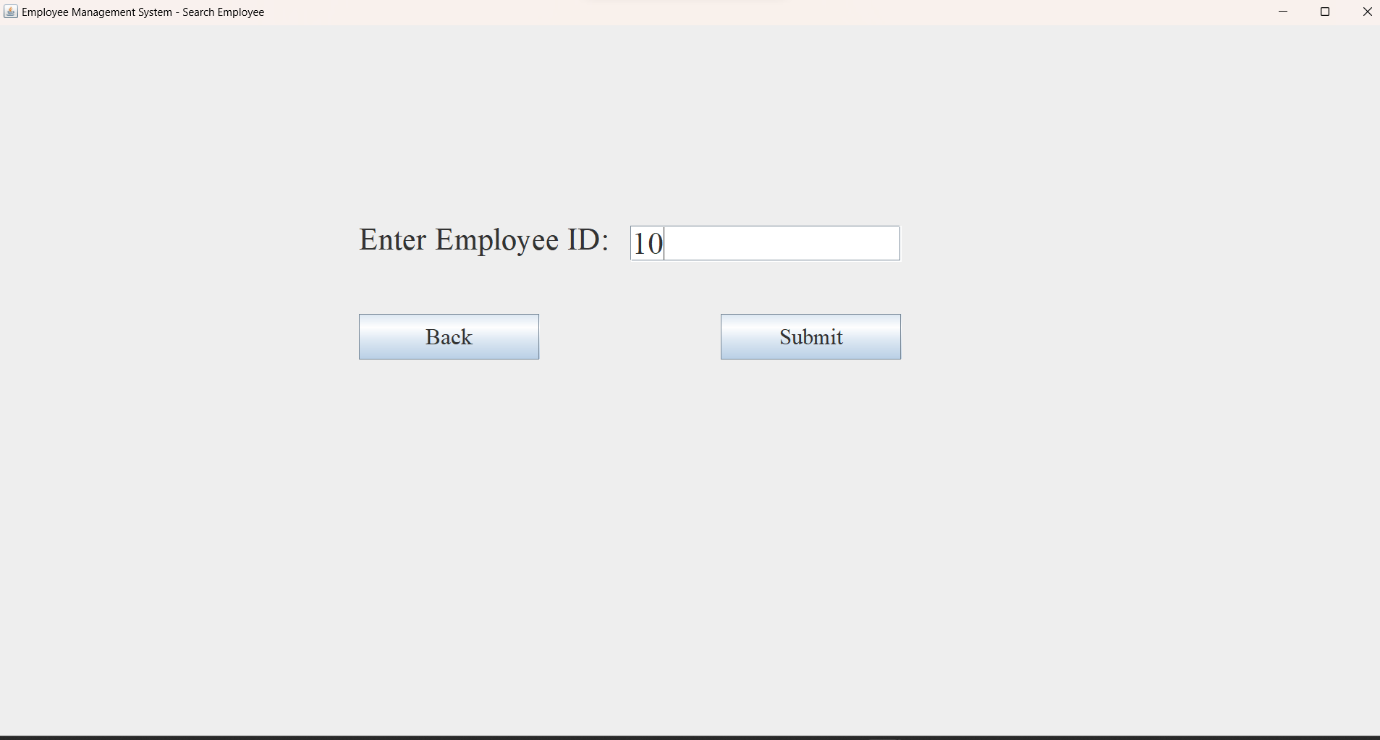
1. **Edit Employee Details – Fields**

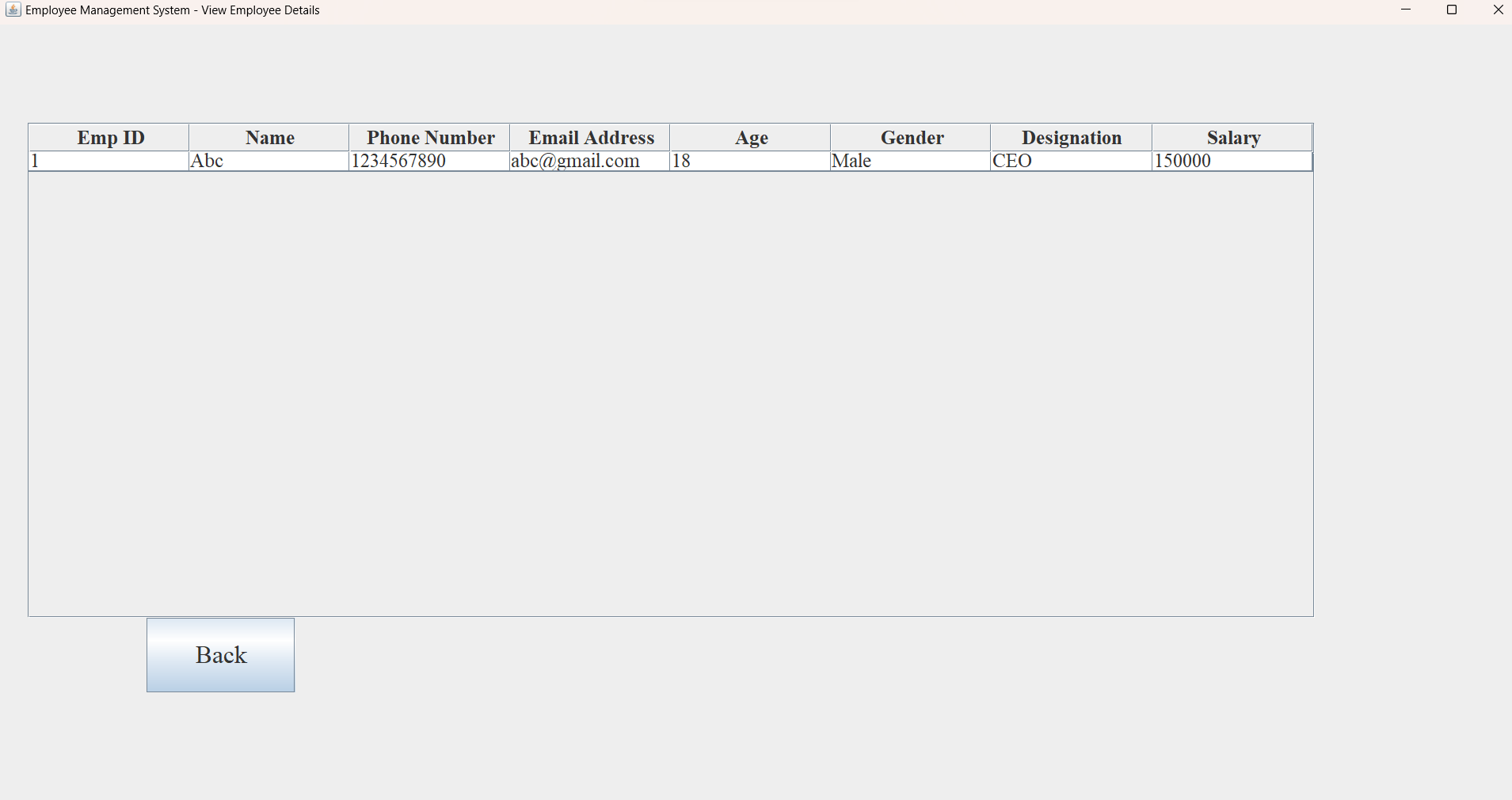
****

1. **Edit Employee Details – Record Updated Successfully**

****

1. **Search Employee Details**

****

****

1. **Skill Developed:**
2. We learned how to define a class and how to create object of class.
3. We learned how to implement the array of object.
4. We learned the concept of menu driven program.
5. We learned how to import package in the program.
6. **Applications of this Micro-Project:**
7. The employee record management system is use in the offices for managing the information of employee.
8. **Area of Future Improvement:**

**Integration with other systems:** For various functions including HR administration, payroll, and time and attendance monitoring, many businesses employee a range of different software systems. These technologies can be integrated with employee management systems to increase productivity and decrease error rates.

**Data analysis and reporting**: Payroll data, attendance records, and performance indicators are just a few of the types of data that employee management systems can produce. Companies can learn a lot about employee performance, trends, and areas for improvement by integrating powerful data analysis and reporting features.

**Security:** Sensitive data, including personal and payroll information, is contained in employee management systems. Security needs to be significantly improved as a result. Systems for data backup and recovery, access control, and strong encryption can all aid in preventing data breaches and safeguarding sensitive data.