

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.

2. Course Outcomes Achieved:

- a) Develop program using GUI framework (AWT and Swing).
- b) Handle events of AWT and Swing Components.
- c) Develop programs using Database.
- d) Develop programs to handle events in Java Programming

3. Literature Review:

3.1 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.

3.2 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.

3.3 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.

3.4 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.

4. 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(des);  
emp_add.add(des_i);  
emp_add.add(des_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:");

```

```

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

```

5. Actual Resources Required:

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

6. Output of Micro-Project:

a) Home Page



b) Add Employee Details

Employee Management System - Add Employee

Enter Emp ID:

Enter Name:

Enter Phone Number:

Enter Email Address:

Enter Age:

Select Gender:

Enter Designation:

Enter Salary (Rs):

c) Add Employee Details - Validation

Employee Management System - Add Employee

Enter Emp ID:
Enter a valid phone number (10 Digits)

Enter Name:
Enter Name in Alphabet

Enter Phone Number:
Enter a valid phone number (10 Digits)

Enter Email Address:
Email should contain '@' and '.'

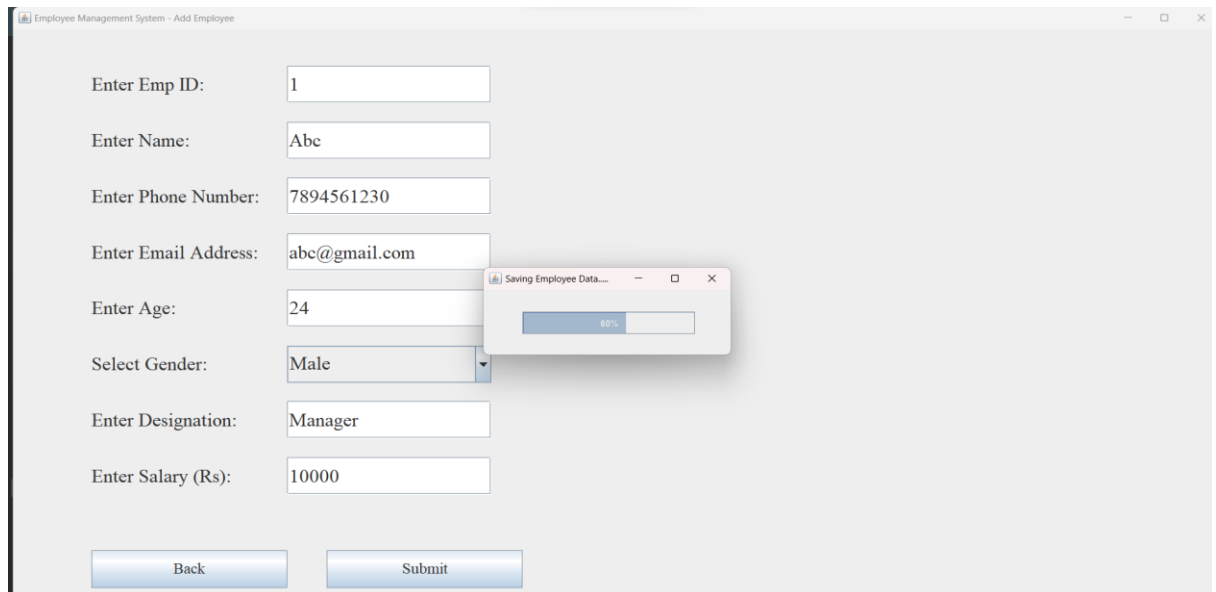
Enter Age:
Enter a valid age (above 18)

Select Gender:
Please Select Your Gender

Enter Designation:
Enter Designation in Alphabet

Enter Salary (Rs):
Salary cannot be a negative number

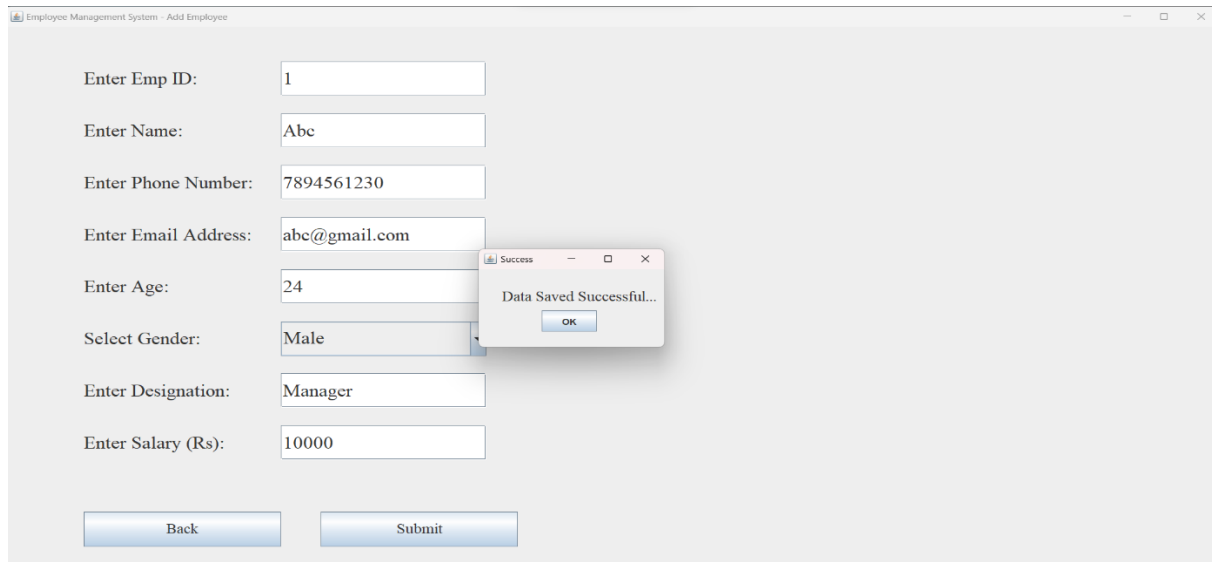
d) Add Employee Details – Loading and Success Message



The screenshot shows a web application window titled "Employee Management System - Add Employee". It contains a form with the following fields and values:

- Enter Emp ID: 1
- Enter Name: Abc
- Enter Phone Number: 7894561230
- Enter Email Address: abc@gmail.com
- Enter Age: 24
- Select Gender: Male (dropdown menu)
- Enter Designation: Manager
- Enter Salary (Rs): 10000

At the bottom of the form are two buttons: "Back" and "Submit". A small modal dialog titled "Saving Employee Data..." is overlaid on the form, showing a progress bar at 60%.



The screenshot shows the same "Add Employee" form as above, but with a different modal dialog. The dialog is titled "Success" and displays the message "Data Saved Successful...". It has an "OK" button.

e) View Employee Details

Employee Management System - View Employee Details

Emp ID	Name	Phone Number	Email Address	Age	Gender	Designation	Salary
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.c...	18	Male	Tester	800000
5	Mohit	7824692159	mohit@gmail.com	18	Male	Designer	750000

Back

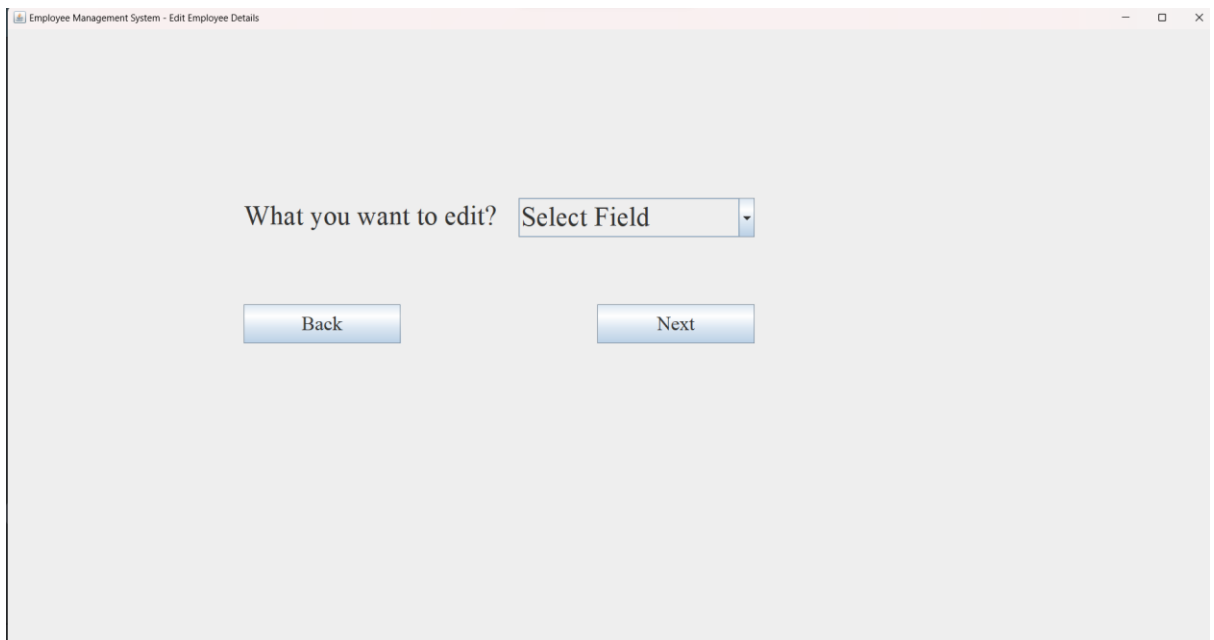
f) Edit Employee Details

Employee Management System - Search Employee

Enter Employee ID:

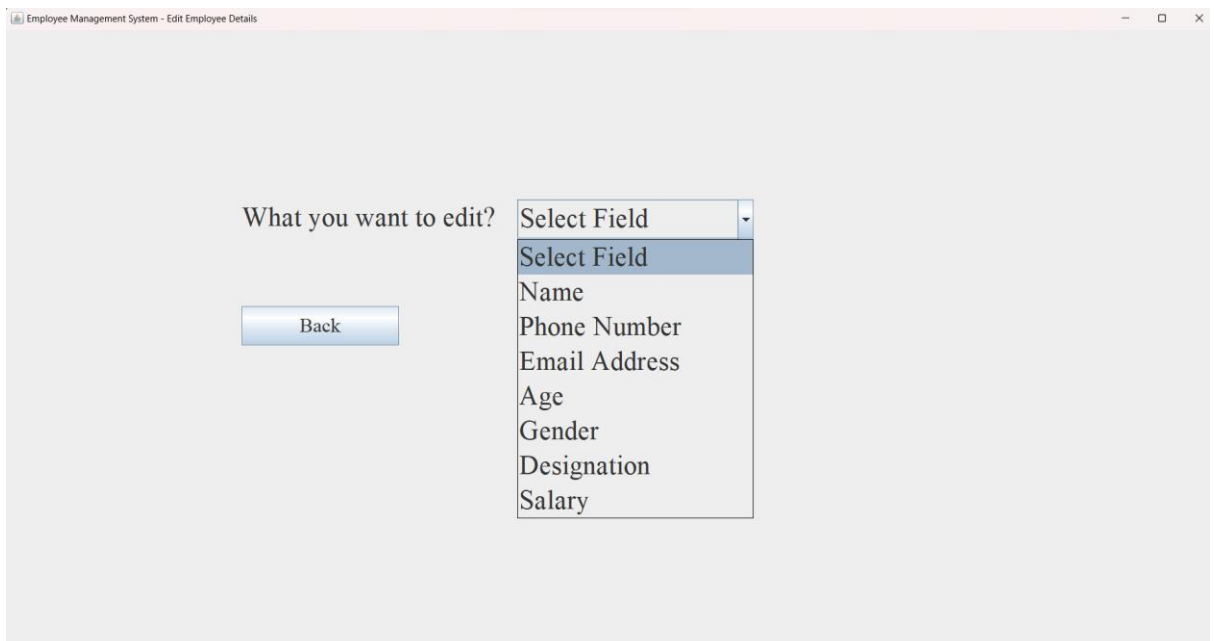
Back Submit

g) Edit Employee Details - Select Field



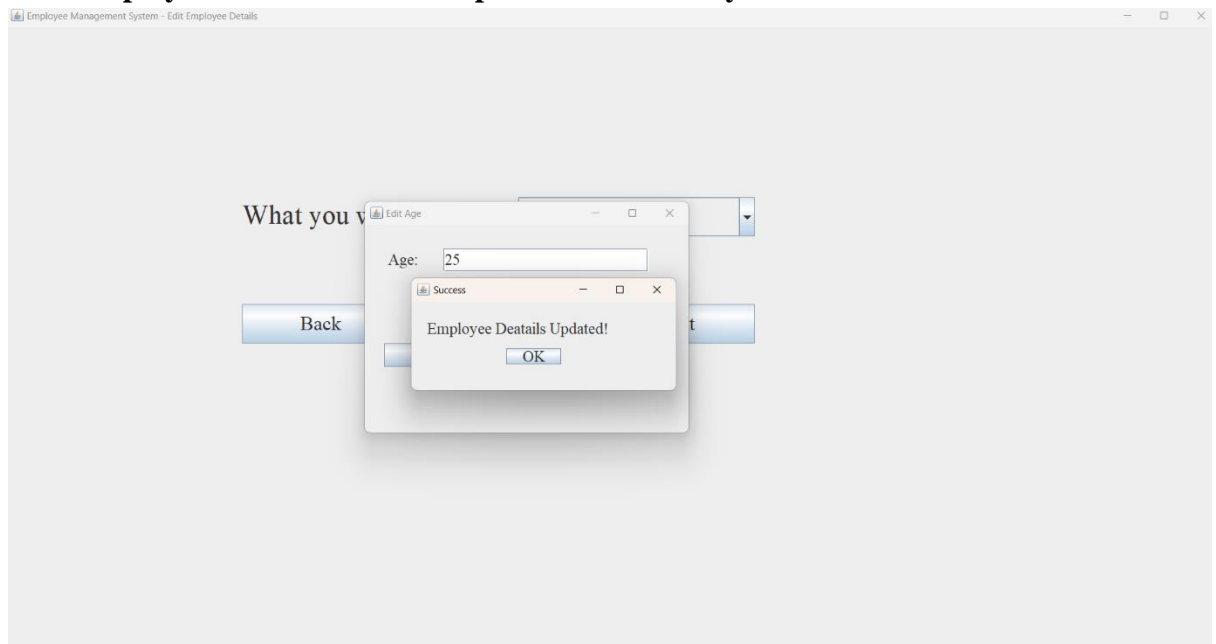
The screenshot shows a window titled "Employee Management System - Edit Employee Details". The main content area has a light gray background. In the center, the text "What you want to edit?" is followed by a dropdown menu labeled "Select Field". Below the dropdown, there are two buttons: "Back" on the left and "Next" on the right. The window has a standard title bar with minimize, maximize, and close buttons.

h) Edit Employee Details – Fields

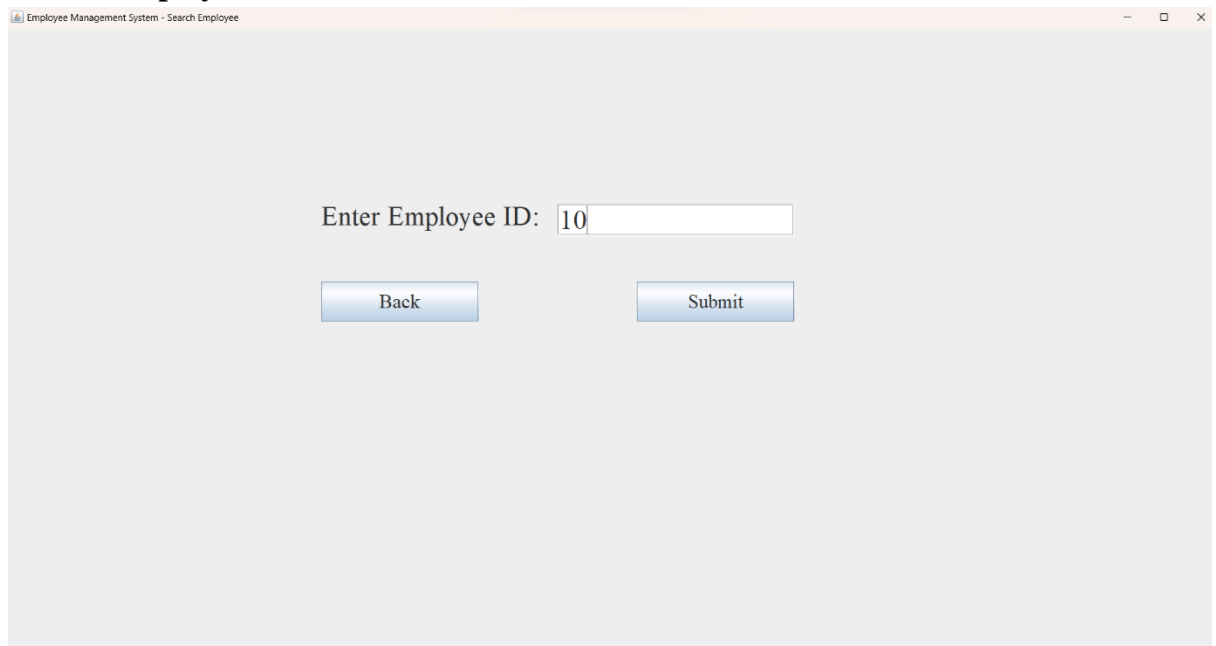


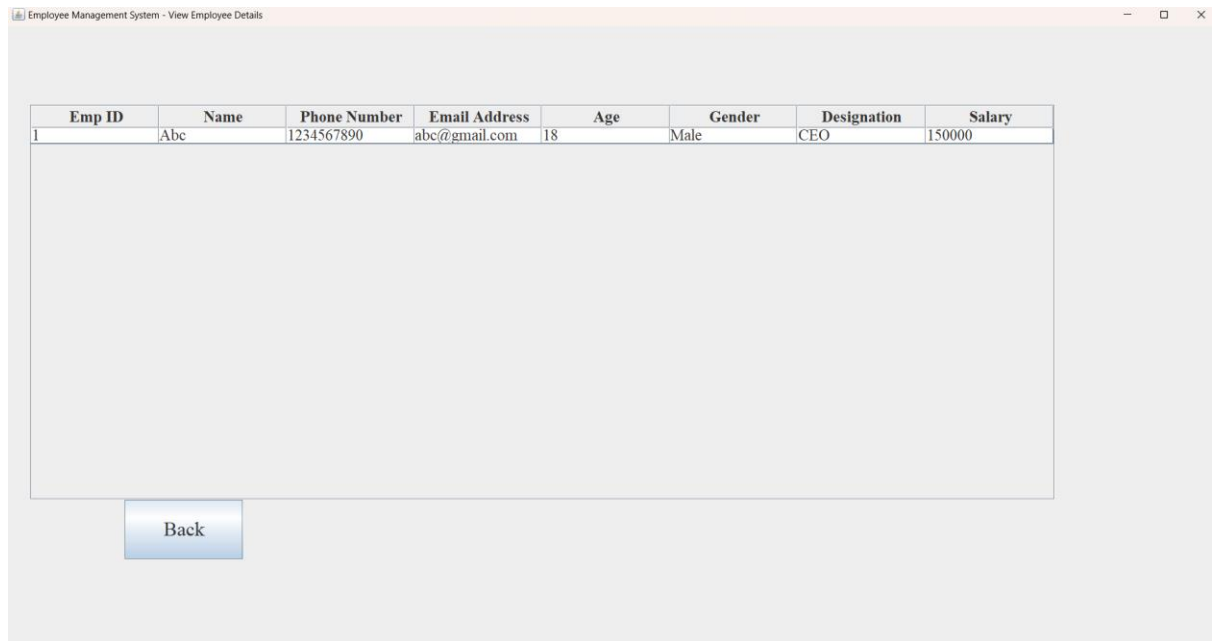
The screenshot shows the same window as in the previous image, but the dropdown menu is now open. The list of fields to edit is displayed below the "Select Field" header. The fields are: Name, Phone Number, Email Address, Age, Gender, Designation, and Salary. The "Back" button is visible on the left side of the window. The window title bar and background are the same as in the previous image.

i) Edit Employee Details – Record Updated Successfully



j) Search Employee Details





7. Skill Developed:

- 1) We learned how to define a class and how to create object of class.
- 2) We learned how to implement the array of object.
- 3) We learned the concept of menu driven program.
- 4) We learned how to import package in the program.

8. Applications of this Micro-Project:

- 1) The employee record management system is use in the offices for managing the information of employee.

9. Area of Future Improvement:

Integration with other systems: For various functions including HR administration, payroll, and time and attendance monitoring, many businesses employ 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.