



Branch of Computer Science and Engineering

Project

Employee Management Application

Submitted by:

Group 5

AP21110011491 - Dhanushree Y

AP21110011516 - Kalyan Ram P

AP21110011536 - Madhuri K

AP21110011548 - Hima Teja G

Course:

Java Programming

Guided by:

Mr. Sudhakar Reddy N

Contents

1. Abstract
2. Source code
3. Output
4. Explanation of the code
5. Conclusion

Abstract

Employee Management Application helps us to maintain the details of employees working in an organization. Earlier people used records to maintain the details of an employee which was comparatively difficult to retrieve, access, or update the data of a person. Also, there was a lack of confidentiality in managing the details and the chances of losing the data were likely high.

Using this application, we can easily create, add, update, access, delete, and view an employee's data as required. Also, it can be accessed only by authorized persons unlike when maintained in records as physical information. The chances of losing the data are insignificant. Thus, the data is confidential and secured.

Source Code

```
Start Page X Login.java X Conn.java X ViewEmployee.java X UpdateEmployee.java X Splash.java X RemoveEmployee.java X Home.java...
Source History

1 package employee.management.system;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.*;
6 import java.sql.*;
7
8 public class Login extends JFrame implements ActionListener{
9
10     JTextField tfusername, tfpassword;
11
12     Login() {
13
14         getContentPane().setBackground(c: Color.WHITE);
15         setLayout(manager: null);
16
17         JLabel lblusername = new JLabel(text: "Username");
18         lblusername.setBounds(x: 40, y: 20, width: 100, height:30);
19         add(comp: lblusername);
20
21         tfusername = new JTextField();
22         tfusername.setBounds(x: 150, y: 20, width: 150, height:30);
23         add(comp: tfusername);
24
25         JLabel lblpassword = new JLabel(text: "Password");
26         lblpassword.setBounds(x: 40, y: 70, width: 100, height:30);
27         add(comp: lblpassword);
28
29         tfpassword = new JTextField();
30         tfpassword.setBounds(x: 150, y: 70, width: 150, height:30);
31         add(comp: tfpassword);
32
33         JButton login = new JButton(text: "LOGIN");
34         login.setBounds(x: 150, y: 140, width: 150, height:30);
35         login.setBackground(bg: Color.BLACK);
36         login.setForeground(fg: Color.WHITE);
37         login.addActionListener(this);
38         add(comp: login);
39
40         ImageIcon i1 = new ImageIcon(location: ClassLoader.getResource(name: "icons/second.jpg"));
41         Image i2 = i1.getImage().getScaledInstance(width: 200, height:200, hints: Image.SCALE_DEFAULT);
42         ImageIcon i3 = new ImageIcon(image: i2);
43         JLabel image = new JLabel(image: i3);
44         image.setBounds(x: 350, y: 0, width: 200, height:200);
45         add(comp: image);
46
47         setSize(width: 600, height:300);
48         setLocation(x: 450, y: 200);
49         setVisible(b: true);
50     }
51
52     public void actionPerformed(ActionEvent ae) {
53         try {
54             String username = tfusername.getText();
55             String password = tfpassword.getText();
56
57             Conn c = new Conn();
58             String query = "select * from login where username = '"+username+"' and password = '"+password+"'";
59
60             ResultSet rs = c.s.executeQuery(string: query);
61             if (rs.next()) {
62                 setVisible(b: false);
63                 new Home();
64             } else {
65                 JOptionPane.showMessageDialog(parentComponent: null, message: "Invalid username or password");
66                 setVisible(b: false);
67             }
68         } catch (Exception e) {
69             e.printStackTrace();
70         }
71     }
72
73     public static void main(String[] args) {
74         new Login();
75     }
76 }
```

```
Start Page X Login.java X Conn.java X ViewEmployee.java X UpdateEmployee.java X Splash.java X RemoveEmployee.java X Home.java... < > v
Source History
1 package employee.management.system;
2
3 import java.sql.*;
4
5 public class Conn {
6
7     Connection c;
8     Statement s;
9
10 public Conn () {
11     try {
12         Class.forName(className:"com.mysql.cj.jdbc.Driver");
13         c = DriverManager.getConnection(url:"jdbc:mysql:///emp", user:"root", password:"1225");
14         s = c.createStatement();
15     } catch (Exception e) {
16         e.printStackTrace();
17     }
18 }
19 }
```

```
Start Page X Login.java X Conn.java X ViewEmployee.java X UpdateEmployee.java X Splash.java X RemoveEmployee.java X Home.java... < > v
Source History
1 package employee.management.system;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.sql.*;
6 import net.proteanit.sql.DbUtils;
7 import java.awt.event.*;
8
9 public class ViewEmployee extends JFrame implements ActionListener{
10
11     JTable table;
12     Choice employeeId;
13     JButton search, print, update, back;
14
15     ViewEmployee() {
16
17         getContentPane().setBackground(c: Color.WHITE);
18         setLayout(manager: null);
19
20         JLabel searchlbl = new JLabel(text: "Search by Employee Id");
21         searchlbl.setBounds(x: 20, y: 20, width: 150, height:20);
22         add(comp: searchlbl);
23
24         employeeId = new Choice();
25         employeeId.setBounds(x: 180, y: 20, width: 150, height:20);
26         add(comp: employeeId);
27         try {
28             Conn c = new Conn();
29             ResultSet rs = c.s.executeQuery(string:"select * from employee");
30             while(rs.next()) {
31                 employeeId.add(item: rs.getString(string:"empId"));
32             }
33         } catch (Exception e) {
34             e.printStackTrace();
35         }
36
37         table = new JTable();
38
39         try {
40             Conn c = new Conn();
41             ResultSet rs = c.s.executeQuery(string:"select * from employee");
42             table.setModel(dataModel:DbUtils.resultSetToTableModel(rs));
43         } catch (Exception e) {
44             e.printStackTrace();
45         }
46
47         JScrollPane jsp = new JScrollPane(view: table);
48         jsp.setBounds(x: 0, y: 100, width: 900, height:600);
49         add(comp: jsp);
50
51         search = new JButton(text: "Search");
52         search.setBounds(x: 20, y: 70, width: 80, height:20);
53         search.addActionListener(l: this);
```

```

55     add(comp: search);
56
57     print = new JButton(text: "Print");
58     print.setBounds(x: 120, y: 70, width: 80, height: 20);
59     print.addActionListener(l: this);
60     add(comp: print);
61
62     update = new JButton(text: "Update");
63     update.setBounds(x: 220, y: 70, width: 80, height: 20);
64     update.addActionListener(l: this);
65     add(comp: update);
66
67     back = new JButton(text: "Back");
68     back.setBounds(x: 320, y: 70, width: 80, height: 20);
69     back.addActionListener(l: this);
70     add(comp: back);
71
72     setSize(width: 900, height: 700);
73     setLocation(x: 300, y: 100);
74     setVisible(b: true);
75 }
76
77 public void actionPerformed(ActionEvent ae) {
78     if (ae.getSource() == search) {
79         String query = "select * from employee where empId = '"+employeeId.getSelectedItem()+"'";
80         try {
81             Conn c = new Conn();
82             ResultSet rs = c.s.executeQuery(string: query);
83             table.setModel(dataModel: DbUtils.resultSetToTableModel(rs));
84         } catch (Exception e) {
85             e.printStackTrace();
86         }
87     } else if (ae.getSource() == print) {
88         try {
89             table.print();
90         } catch (Exception e) {
91             e.printStackTrace();
92         }
93     } else if (ae.getSource() == update) {
94         setVisible(b: false);
95         new UpdateEmployee(empId: employeeId.getSelectedItem());
96     } else {
97         setVisible(b: false);
98         new Home();
99     }
100 }
101
102 public static void main(String[] args) {
103     new ViewEmployee();
104 }
105 }

```

```
Start Page X Login.java X Conn.java X ViewEmployee.java X UpdateEmployee.java X Splash.java X RemoveEmployee.java X Homej... < > V
Source History
1 package employee.management.system;
2
3 import java.awt.*;
4 import javax.swing.*;
5 import java.awt.event.*;
6 import java.sql.*;
7
8 public class UpdateEmployee extends JFrame implements ActionListener{
9
10     JTextField tfeducation, tfname, tfaddress, tfphone, tfaadhar, tfemail, tfsalary, tfdesignation;
11     JLabel lblempId;
12     JButton add, back;
13     String empId;
14
15     UpdateEmployee(String empId) {
16         this.empId = empId;
17         getContentPane().setBackground(c: Color.WHITE);
18         setLayout(manager: null);
19
20         JLabel heading = new JLabel(text: "Update Employee Detail");
21         heading.setBounds(x: 320, y: 30, width: 500, height: 50);
22         heading.setFont(new Font(name: "SAN_SERIF", style: Font.BOLD, size: 25));
23         add(comp: heading);
24
25         JLabel labelname = new JLabel(text: "Name");
26         labelname.setBounds(x: 50, y: 150, width: 150, height: 30);
27
28         labelname.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
29         add(comp: labelname);
30
31         JLabel lblname = new JLabel();
32         lblname.setBounds(x: 200, y: 150, width: 150, height: 30);
33         add(comp: lblname);
34
35         JLabel labelfname = new JLabel(text: "Father's Name");
36         labelfname.setBounds(x: 400, y: 150, width: 150, height: 30);
37         labelfname.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
38         add(comp: labelfname);
39
40         tfname = new JTextField();
41         tfname.setBounds(x: 600, y: 150, width: 150, height: 30);
42         add(comp: tfname);
43
44         JLabel labeldob = new JLabel(text: "Date of Birth");
45         labeldob.setBounds(x: 50, y: 200, width: 150, height: 30);
46         labeldob.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
47         add(comp: labeldob);
48
49         JLabel lbldob = new JLabel();
50         lbldob.setBounds(x: 200, y: 200, width: 150, height: 30);
51         add(comp: lbldob);
52
53         JLabel labelsalary = new JLabel(text: "Salary");
54         labelsalary.setBounds(x: 400, y: 200, width: 150, height: 30);
55         add(comp: labelsalary);
56
57         JButton add = new JButton(text: "Add");
58         add.setBounds(x: 600, y: 250, width: 100, height: 30);
59         add.addActionListener(this);
60
61         JButton back = new JButton(text: "Back");
62         back.setBounds(x: 700, y: 250, width: 100, height: 30);
63         back.addActionListener(this);
64
65         setTitle("Update Employee");
66         setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
67         setSize(800, 600);
68         setLocation(300, 100);
69         setVisible(true);
70     }
71
72     public void actionPerformed(ActionEvent ae) {
73         if (ae.getSource() == add) {
74             // Add logic here
75         }
76         if (ae.getSource() == back) {
77             // Back logic here
78         }
79     }
80 }
```

```

53 labelsalary.setBounds(x: 400, y: 200, width: 150, height: 30);
54 labelsalary.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
55 add(comp: labelsalary);
56
57 tfsalary = new JTextField();
58 tfsalary.setBounds(x: 600, y: 200, width: 150, height: 30);
59 add(comp: tfsalary);
60
61 JLabel labeladdress = new JLabel(text: "Address");
62 labeladdress.setBounds(x: 50, y: 250, width: 150, height: 30);
63 labeladdress.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
64 add(comp: labeladdress);
65
66 tfaddress = new JTextField();
67 tfaddress.setBounds(x: 200, y: 250, width: 150, height: 30);
68 add(comp: tfaddress);
69
70 JLabel labelphone = new JLabel(text: "Phone");
71 labelphone.setBounds(x: 400, y: 250, width: 150, height: 30);
72 labelphone.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
73 add(comp: labelphone);
74
75 tfphone = new JTextField();
76 tfphone.setBounds(x: 600, y: 250, width: 150, height: 30);
77 add(comp: tfphone);
78
79 JLabel lalemail = new JLabel(text: "Email");
80 lalemail.setBounds(x: 50, y: 300, width: 150, height: 30);
81 lalemail.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
82 add(comp: lalemail);
83
84 tfemail = new JTextField();
85 tfemail.setBounds(x: 200, y: 300, width: 150, height: 30);
86 add(comp: tfemail);
87
88 JLabel labelededucation = new JLabel(text: "Higest Education");
89 labelededucation.setBounds(x: 400, y: 300, width: 150, height: 30);
90 labelededucation.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
91 add(comp: labelededucation);
92
93 tfeducation = new JTextField();
94 tfeducation.setBounds(x: 600, y: 300, width: 150, height: 30);
95 add(comp: tfeducation);
96
97 JLabel labeldesignation = new JLabel(text: "Designation");
98 labeldesignation.setBounds(x: 50, y: 350, width: 150, height: 30);
99 labeldesignation.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
100 add(comp: labeldesignation);
101
102 tfdesignation = new JTextField();
103 tfdesignation.setBounds(x: 200, y: 350, width: 150, height: 30);
104 add(comp: tfdesignation);
105
106 JLabel labelaadhar = new JLabel(text: "Aadhar Number");
107 labelaadhar.setBounds(x: 400, y: 350, width: 150, height: 30);
108 labelaadhar.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
109 add(comp: labelaadhar);
110
111 JLabel lblaadhar = new JLabel();
112 lblaadhar.setBounds(x: 600, y: 350, width: 150, height: 30);
113 add(comp: lblaadhar);
114
115 JLabel labelempId = new JLabel(text: "Employee id");
116 labelempId.setBounds(x: 50, y: 400, width: 150, height: 30);
117 labelempId.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
118 add(comp: labelempId);
119
120 lblempId = new JLabel();
121 lblempId.setBounds(x: 200, y: 400, width: 150, height: 30);
122 lblempId.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
123 add(comp: lblempId);
124
125 try {
126     Conn c = new Conn();
127     String query = "select * from employee where empId = '"+empId+"'";
128     ResultSet rs = c.s.executeQuery(string: query);
129     while(rs.next()) {
130         lblname.setText(text: rs.getString(string: "name"));
131         tffname.setText(t: rs.getString(string: "fname"));

```



```

132         lbldob.setText(text: rs.getString(string: "dob"));
133         tfaddress.setText(t: rs.getString(string: "address"));
134         tfsalary.setText(t: rs.getString(string: "salary"));
135         tfphone.setText(t: rs.getString(string: "phone"));
136         tfemail.setText(t: rs.getString(string: "email"));
137         tfeducation.setText(t: rs.getString(string: "education"));
138         lblaadhar.setText(text: rs.getString(string: "aadhar"));
139         lblempId.setText(text: rs.getString(string: "empId"));
140         tfdesignation.setText(t: rs.getString(string: "designation"));
141     }
142 }
143 } catch (Exception e) {
144     e.printStackTrace();
145 }
146
147 add = new JButton(text: "Update Details");
148 add.setBounds(x: 250, y: 550, width: 150, height: 40);
149 add.addActionListener(i: this);
150 add.setBackground(bg: Color.BLACK);
151 add.setForeground(fg: Color.WHITE);
152 add(comp: add);
153
154 back = new JButton(text: "Back");
155 back.setBounds(x: 450, y: 550, width: 150, height: 40);
156 back.addActionListener(i: this);
157 back.setBackground(bg: Color.BLACK);
158 back.setForeground(fg: Color.WHITE);
159 add(comp: back);
160
161 setSize(width: 900, height: 700);
162 setLocation(x: 300, y: 50);
163 setVisible(b: true);
164 }
165
166 public void actionPerformed(ActionEvent ae) {
167     if (ae.getSource() == add) {
168         String fname = tffname.getText();
169         String salary = tfsalary.getText();
170         String address = tfaddress.getText();
171         String phone = tfphone.getText();
172         String email = tfemail.getText();
173         String education = tfeducation.getText();
174         String designation = tfdesignation.getText();
175
176         try {
177             Conn conn = new Conn();
178             String query = "update employee set fname = '"+fname+"', salary = '"+salary+"', address = '"+address+"', phone
179             conn.s.executeUpdate(string: query);
180             JOptionPane.showMessageDialog(parentComponent: null, message: "Details updated successfully");
181             setVisible(b: false);
182             new Home();
183         } catch (Exception e) {
184             e.printStackTrace();
185         }
186     } else {
187         setVisible(b: false);
188         new Home();
189     }
190 }
191
192 public static void main(String[] args) {
193     new UpdateEmployee(empId: "");
194 }
195 }
196

```

```

Start Page x Login.java x Conn.java x ViewEmployee.java x UpdateEmployee.java x Splash.java x RemoveEmployee.java x Home.java...
Source History
1 package employee.management.system;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.*;
6
7 public class Splash extends JFrame implements ActionListener {
8
9     Splash() {
10
11         getContentPane().setBackground(c: Color.WHITE);
12         setLayout(manager: null);
13
14         JLabel heading = new JLabel(text: "EMPLOYEE MANAGEMENT SYSTEM");
15         heading.setBounds(x: 80, y: 30, width: 1200, height: 60);
16         heading.setFont(new Font(name: "serif", style: Font.PLAIN, size: 60));
17         heading.setForeground(fg: Color.RED);
18         add(comp: heading);
19
20         ImageIcon i1 = new ImageIcon(location: ClassLoader.getResource(name: "icons/front.jpg"));
21         Image i2 = i1.getImage().getScaledInstance(width: 1100, height: 700, hints: Image.SCALE_DEFAULT);
22         ImageIcon i3 = new ImageIcon(image: i2);
23         JLabel image = new JLabel(image: i3);
24         image.setBounds(x: 50, y: 100, width: 1050, height: 500);
25         add(comp: image);
26
27         JButton clickhere = new JButton(text: "CLICK HERE TO CONTINUE");
28         clickhere.setBounds(x: 400, y: 400, width: 300, height: 70);
29         clickhere.setBackground(bg: Color.BLACK);
30         clickhere.setForeground(fg: Color.WHITE);
31         clickhere.addActionListener(l: this);
32         image.add(comp: clickhere);
33
34
35         setSize(width: 1170, height: 650);
36         setLocation(x: 200, y: 50);
37         setVisible(b: true);
38
39         while(true) {
40             heading.setVisible(aFlag: false);
41             try {
42                 Thread.sleep(millis: 500);
43             } catch (Exception e) {
44
45             }
46
47             heading.setVisible(aFlag: true);
48             try {
49                 Thread.sleep(millis: 500);
50             } catch (Exception e) {
51
52             }
53         }
54     }
55
56     public void actionPerformed(ActionEvent ae) {
57         setVisible(b: false);
58         new Login();
59     }
60
61     public static void main(String args[]) {
62         new Splash();
63     }
64 }

```

```
Start Page x Login.java x Conn.java x ViewEmployee.java x UpdateEmployee.java x Splash.java x RemoveEmployee.java x Home.java...
Source History

1 package employee.management.system;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.sql.*;
6 import java.awt.event.*;
7
8 public class RemoveEmployee extends JFrame implements ActionListener {
9
10     Choice cEmpId;
11     JButton delete, back;
12
13     RemoveEmployee() {
14         getContentPane().setBackground(c: Color.WHITE);
15         setLayout(manager: null);
16
17         JLabel labelEmpId = new JLabel(text: "Employee Id");
18         labelEmpId.setBounds(x: 50, y: 50, width: 100, height: 30);
19         add(comp: labelEmpId);
20
21         cEmpId = new Choice();
22         cEmpId.setBounds(x: 200, y: 50, width: 150, height: 30);
23         add(comp: cEmpId);
24
25         try {
26             Conn c = new Conn();
27             String query = "select * from employee";
28             ResultSet rs = c.s.executeQuery(string: query);
29             while(rs.next()) {
30                 cEmpId.add(item: rs.getString(string: "empId"));
31             }
32         } catch (Exception e) {
33             e.printStackTrace();
34         }
35
36         JLabel labelname = new JLabel(text: "Name");
37         labelname.setBounds(x: 50, y: 100, width: 100, height: 30);
38         add(comp: labelname);
39
40         JLabel lblname = new JLabel();
41         lblname.setBounds(x: 200, y: 100, width: 100, height: 30);
42         add(comp: lblname);
43
44         JLabel labelphone = new JLabel(text: "Phone");
45         labelphone.setBounds(x: 50, y: 150, width: 100, height: 30);
46         add(comp: labelphone);
47
48         JLabel lblphone = new JLabel();
49         lblphone.setBounds(x: 200, y: 150, width: 100, height: 30);
50         add(comp: lblphone);
51
52         JLabel labelemail = new JLabel(text: "Email");
53         labelemail.setBounds(x: 50, y: 200, width: 100, height: 30);
54         add(comp: labelemail);
55
56         JLabel lblemail = new JLabel();
57         lblemail.setBounds(x: 200, y: 200, width: 100, height: 30);
58         add(comp: lblemail);
59
60         try {
61             Conn c = new Conn();
62             String query = "select * from employee where empId = '"+cEmpId.getSelectedItem()+"'";
63             ResultSet rs = c.s.executeQuery(string: query);
64             while(rs.next()) {
65                 lblname.setText(text: rs.getString(string: "name"));
66                 lblphone.setText(text: rs.getString(string: "phone"));
67                 lblemail.setText(text: rs.getString(string: "email"));
68             }
69         } catch (Exception e) {
70             e.printStackTrace();
71         }
72
73         cEmpId.addItemListener(new ItemListener() {
74             public void itemStateChanged(ItemEvent ie) {
75                 try {
76                     Conn c = new Conn();
77                     String query = "select * from employee where empId = '"+cEmpId.getSelectedItem()+"'";
78                     ResultSet rs = c.s.executeQuery(string: query);
```

```

79         while(rs.next()) {
80             lblname.setText(text: rs.getString(string: "name"));
81             lblphone.setText(text: rs.getString(string: "phone"));
82             lblemail.setText(text: rs.getString(string: "email"));
83         }
84     } catch (Exception e) {
85         e.printStackTrace();
86     }
87 }
88 });
89
90 delete = new JButton(text: "Delete");
91 delete.setBounds(x: 80, y: 300, width: 100,height:30);
92 delete.setBackground(bg: Color.BLACK);
93 delete.setForeground(fg: Color.WHITE);
94 delete.addActionListener(l: this);
95 add(comp: delete);
96
97 back = new JButton(text: "Back");
98 back.setBounds(x: 220, y: 300, width: 100,height:30);
99 back.setBackground(bg: Color.BLACK);
100 back.setForeground(fg: Color.WHITE);
101 back.addActionListener(l: this);
102 add(comp: back);
103
104 ImageIcon i1 = new ImageIcon(location: ClassLoader.getResource(name: "icons/delete.png"));
105 Image i2 = i1.getImage().getScaledInstance(width: 600, height:400, hints: Image.SCALE_DEFAULT);
106 ImageIcon i3 = new ImageIcon(image: i2);
107 JLabel image = new JLabel(image: i3);
108 image.setBounds(x: 350, y: 0, width: 600, height:400);
109 add(comp: image);
110
111 setSize(width: 1000, height:400);
112 setLocation(x: 300, y: 150);
113 setVisible(b: true);
114 }
115
116 public void actionPerformed(ActionEvent ae) {
117     if (ae.getSource() == delete) {
118         try {
119             Conn c = new Conn();
120             String query = "delete from employee where empId = '"+cEmpId.getSelectedItem()+"'";
121             c.s.executeUpdate(string: query);
122             JOptionPane.showMessageDialog(parentComponent: null, message: "Employee Information Deleted Sucessfully");
123             setVisible(b: false);
124             new Home();
125         } catch (Exception e) {
126             e.printStackTrace();
127         }
128     } else {
129         setVisible(b: false);
130         new Home();
131     }
132 }
133
134 public static void main(String[] args) {
135     new RemoveEmployee();
136 }
137 }

```

```

...ge Login.java X Conn.java X ViewEmployee.java X UpdateEmployee.java X Splash.java X RemoveEmployee.java X Home.java X AddEm... < >
Source History
1 package employee.management.system;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.*;
6
7 public class Home extends JFrame implements ActionListener{
8
9     JButton view, add, update, remove;
10
11     Home() {
12
13         setLayout(manager: null);
14
15         ImageIcon i1 = new ImageIcon(location: ClassLoader.getResource(name: "icons/home.jpg"));
16         Image i2 = i1.getImage().getScaledInstance(width: 1120, height: 630, hints: Image.SCALE_DEFAULT);
17         ImageIcon i3 = new ImageIcon(image: i2);
18         JLabel image = new JLabel(image: i3);
19         image.setBounds(x: 0, y: 0, width: 1120, height: 630);
20         add(comp: image);
21
22         JLabel heading = new JLabel(text: "Employee Management System");
23         heading.setBounds(x: 620, y: 20, width: 400, height: 40);
24         heading.setFont(new Font(name: "Raleway", style: Font.BOLD, size: 25));
25         image.add(comp: heading);
26
27         add = new JButton(text: "Add Employee");
28         add.setBounds(x: 650, y: 80, width: 150, height: 40);
29         add.addActionListener(l: this);
30         image.add(comp: add);
31
32         view = new JButton(text: "View Employees");
33         view.setBounds(x: 820, y: 80, width: 150, height: 40);
34         view.addActionListener(l: this);
35         image.add(comp: view);
36
37         update = new JButton(text: "Update Employee");
38         update.setBounds(x: 650, y: 140, width: 150, height: 40);
39         update.addActionListener(l: this);
40         image.add(comp: update);
41
42         remove = new JButton(text: "Remove Employee");
43         remove.setBounds(x: 820, y: 140, width: 150, height: 40);
44         remove.addActionListener(l: this);
45         image.add(comp: remove);
46
47         setSize(width: 1120, height: 630);
48         setLocation(x: 250, y: 100);
49         setVisible(b: true);
50     }
51
52     public void actionPerformed(ActionEvent ae) {
53         if (ae.getSource() == add) {
54             setVisible(b: false);
55             new AddEmployee();
56         } else if (ae.getSource() == view) {
57             setVisible(b: false);
58             new ViewEmployee();
59         } else if (ae.getSource() == update) {
60             setVisible(b: false);
61             new ViewEmployee();
62         } else {
63             setVisible(b: false);
64             new RemoveEmployee();
65         }
66     }
67
68     public static void main(String[] args) {
69         new Home();
70     }
71 }

```

```
...va Conn.java x ViewEmployee.java x UpdateEmployee.java x Splash.java x RemoveEmployee.java x Home.java x AddEmployee.java x
Source History
1 package employee.management.system;
2
3 import java.awt.*;
4 import javax.swing.*;
5 import com.toedter.calendar.JDateChooser;
6 import java.util.*;
7 import java.awt.event.*;
8
9 public class AddEmployee extends JFrame implements ActionListener{
10
11     Random ran = new Random();
12     int number = ran.nextInt(bound: 999999);
13
14     JTextField tfname, tfaddress, tfphone, tfaadhar, tfemail, tfsalary, tfdesignation;
15     JDateChooser dcdob;
16     JComboBox cbeducation;
17     JLabel lblempId;
18     JButton add, back;
19
20     AddEmployee() {
21         getContentPane().setBackground(c: Color.WHITE);
22         setLayout(manager: null);
23
24         JLabel heading = new JLabel(text: "Add Employee Detail");
25         heading.setBounds(x: 320, y: 30, width: 500, height: 50);
26         heading.setFont(new Font(name: "SAN_SERIF", style: Font.BOLD, size: 25));
27
28         add(comp: heading);
29
30         JLabel labelname = new JLabel(text: "Name");
31         labelname.setBounds(x: 50, y: 150, width: 150, height: 30);
32         labelname.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
33         add(comp: labelname);
34
35         tfname = new JTextField();
36         tfname.setBounds(x: 200, y: 150, width: 150, height: 30);
37         add(comp: tfname);
38
39         JLabel labelfname = new JLabel(text: "Father's Name");
40         labelfname.setBounds(x: 400, y: 150, width: 150, height: 30);
41         labelfname.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
42         add(comp: labelfname);
43
44         tffname = new JTextField();
45         tffname.setBounds(x: 600, y: 150, width: 150, height: 30);
46         add(comp: tffname);
47
48         JLabel labeldob = new JLabel(text: "Date of Birth");
49         labeldob.setBounds(x: 50, y: 200, width: 150, height: 30);
50         labeldob.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
51         add(comp: labeldob);
52
53         dcdob = new JDateChooser();
54         dcdob.setBounds(x: 200, y: 200, width: 150, height: 30);
55         add(comp: dcdob);
56
57         JLabel labelsalary = new JLabel(text: "Salary");
58         labelsalary.setBounds(x: 400, y: 200, width: 150, height: 30);
59         labelsalary.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
60         add(comp: labelsalary);
61
62         tfsalary = new JTextField();
63         tfsalary.setBounds(x: 600, y: 200, width: 150, height: 30);
64         add(comp: tfsalary);
65
66         JLabel labeladdress = new JLabel(text: "Address");
67         labeladdress.setBounds(x: 50, y: 250, width: 150, height: 30);
68         labeladdress.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
69         add(comp: labeladdress);
70
71         tfaddress = new JTextField();
72         tfaddress.setBounds(x: 200, y: 250, width: 150, height: 30);
73         add(comp: tfaddress);
74
75         JLabel labelphone = new JLabel(text: "Phone");
76         labelphone.setBounds(x: 400, y: 250, width: 150, height: 30);
77         labelphone.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
78         add(comp: labelphone);
79
80         tfphone = new JTextField();
81         tfphone.setBounds(x: 600, y: 250, width: 150, height: 30);
```

```

81      add(comp: tfphone);
82
83      JLabel lalelemail = new JLabel(text: "Email");
84      lalelemail.setBounds(x: 50, y: 300, width: 150, height: 30);
85      lalelemail.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
86      add(comp: lalelemail);
87
88      tfemail = new JTextField();
89      tfemail.setBounds(x: 200, y: 300, width: 150, height: 30);
90      add(comp: tfemail);
91
92      JLabel labeleducation = new JLabel(text: "Highest Education");
93      labeleducation.setBounds(x: 400, y: 300, width: 150, height: 30);
94      labeleducation.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
95      add(comp: labeleducation);
96
97      String courses[] = {"BBA", "BCA", "BA", "BSC", "B.COM", "BTech", "MBA", "MCA", "MA", "MTech", "MSC", "PHD"};
98      cbeducation = new JComboBox(items: courses);
99      cbeducation.setBackground(bg: Color.WHITE);
100     cbeducation.setBounds(x: 600, y: 300, width: 150, height: 30);
101     add(comp: cbeducation);
102
103     JLabel labeldesignation = new JLabel(text: "Designation");
104     labeldesignation.setBounds(x: 50, y: 350, width: 150, height: 30);
105     labeldesignation.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
106     add(comp: labeldesignation);

```

```

108     tfdesignation = new JTextField();
109     tfdesignation.setBounds(x: 200, y: 350, width: 150, height: 30);
110     add(comp: tfdesignation);
111
112     JLabel labelaadhar = new JLabel(text: "Aadhar Number");
113     labelaadhar.setBounds(x: 400, y: 350, width: 150, height: 30);
114     labelaadhar.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
115     add(comp: labelaadhar);
116
117     tfaadhar = new JTextField();
118     tfaadhar.setBounds(x: 600, y: 350, width: 150, height: 30);
119     add(comp: tfaadhar);
120
121     JLabel labelempId = new JLabel(text: "Employee id");
122     labelempId.setBounds(x: 50, y: 400, width: 150, height: 30);
123     labelempId.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
124     add(comp: labelempId);
125
126     lblempId = new JLabel("" + number);
127     lblempId.setBounds(x: 200, y: 400, width: 150, height: 30);
128     lblempId.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
129     add(comp: lblempId);
130
131     add = new JButton(text: "Add Details");
132     add.setBounds(x: 250, y: 550, width: 150, height: 40);
133     add.addActionListener(l: this);

```

```

134     add.setBackground(bg: Color.BLACK);
135     add.setForeground(fg: Color.WHITE);
136     add(comp: add);
137
138     back = new JButton(text: "Back");
139     back.setBounds(x: 450, y: 550, width: 150, height: 40);
140     back.addActionListener(l: this);
141     back.setBackground(bg: Color.BLACK);
142     back.setForeground(fg: Color.WHITE);
143     add(comp: back);
144
145     setSize(width: 900, height: 700);
146     setLocation(x: 300, y: 50);
147     setVisible(b: true);
148 }
149
150 public void actionPerformed(ActionEvent ae) {
151     if (ae.getSource() == add) {
152         String name = tfname.getText();
153         String fname = tffname.getText();
154         String dob = ((JTextField) dcdob.getDateEditor().getUiComponent()).getText();
155         String salary = tfsalary.getText();
156         String address = tfaddress.getText();
157         String phone = tfphone.getText();
158         String email = tfemail.getText();
159         String education = (String) cbeducation.getSelectedItem();
160         String designation = tfdesignation.getText();

```

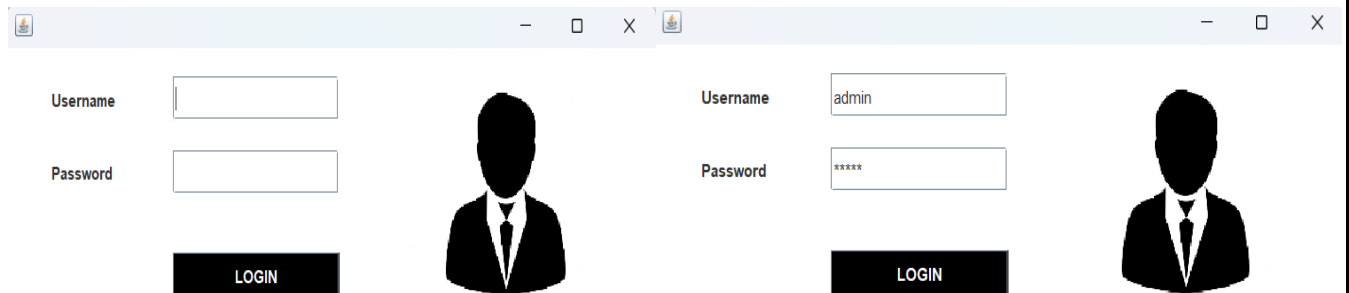
```

161 String aadhar = tfaadhar.getText();
162 String empId = lblEmpId.getText();
163
164 try {
165     Conn conn = new Conn();
166     String query = "insert into employee values('"+name+"', '"+fname+"', '"+dob+"', '"+salary+"', '"+address+"', '";
167     conn.s.executeUpdate(string: query);
168     JOptionPane.showMessageDialog(parentComponent: null, message: "Details added successfully");
169     setVisible(b: false);
170     new Home();
171 } catch (Exception e) {
172     e.printStackTrace();
173 }
174 } else {
175     setVisible(b: false);
176     new Home();
177 }
178 }
179
180 public static void main(String[] args) {
181     new AddEmployee();
182 }
183 }

```


Output

Home Page:

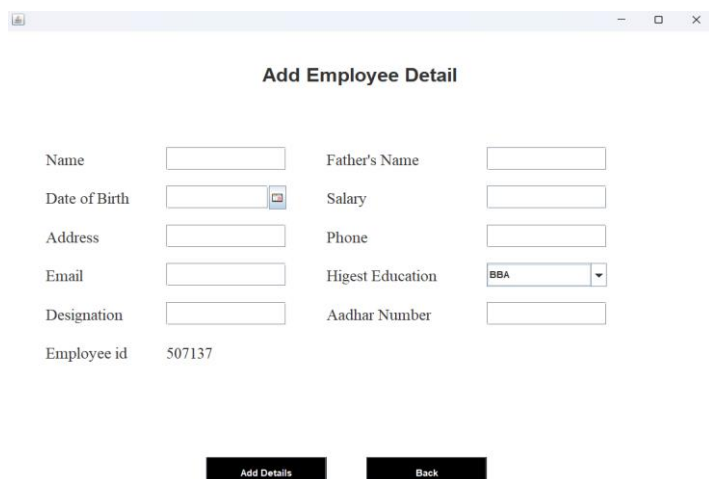


Two browser window screenshots of the Home Page. The left window shows empty login fields (Username, Password) and a LOGIN button, with a silhouette of a person in a suit. The right window shows the Username field filled with 'admin' and the Password field filled with '*****', with the same LOGIN button and silhouette.

Operations:




Add Employee:



Screenshot of the 'Add Employee Detail' form. It contains input fields for Name, Date of Birth, Address, Email, Designation, Father's Name, Salary, Phone, Highest Education (with a dropdown menu showing 'BBA'), and Aadhar Number. The Employee id is pre-filled as 507137. At the bottom are 'Add Details' and 'Back' buttons.


View Employee:

 - □ ×

Search by Employee Id

name	fname	dob	salary	address	phone	email	education	designation	aadhar	empld
Hima	Ram	17-Oct-2003	5000000	vjw	1234567890	abc	BTech	Manager	1234	173681
Dhanush	Shyam	15-May-2004	5000000	def	1532649764	fhsbvhlr	MBA	Senior Analyst	8468468	122267
Madhuri	Arun	18-May-2003	5000000	ksrtbgslitdr	6846854689	dlfbglieue	B.COM	Buisiness An...	6816848	683941
Kalyan	Hrithik	22-Aug-2004	5000000	irubgtiebtuw	6846855225	lrtgblrstb	MCA	Data Analyst	684685165	880861

Update Employee:

 - □ ×

Update Employee Detail

Name

Dhanush

Father's Name

Radhe Shyam

Date of Birth

15-May-2004

Salary

5000000

Address

def

Phone

1532649764

Email

fhsbvhlr

Higest Education

MBA

Designation

Senior Analyst


Aadhar Number

8468468

Employee id

122267

Successful Updation:

 - □ ×

Search by Employee Id

name	fname	dob	salary	address	phone	email	education	designation	aadhar	empld
Hima	Ram	17-Oct-2003	5000000	vjw	1234567890	abc	BTech	Manager	1234	173681
Dhanush	Radhe Shyam	15-May-2004	5000000	def	1532649764	fhsbvhlr	MBA	Senior Analyst	8468468	122267
Madhuri	Arun	18-May-2003	5000000	ksrtbgslitdr	6846854689	dlfbglieue	B.COM	Buisiness An...	6816848	683941
Kalyan	Hrithik	22-Aug-2004	5000000	irubgtiebtuw	6846855225	lrtgblrstb	MCA	Data Analyst	684685165	880861

Delete Employee:

Employee Id

173681

Name

Hima

Phone


1234567890

Email

abc

Delete

Back



Employee Id

173681

Name

Hima

Phone

1234567890

Email

abc

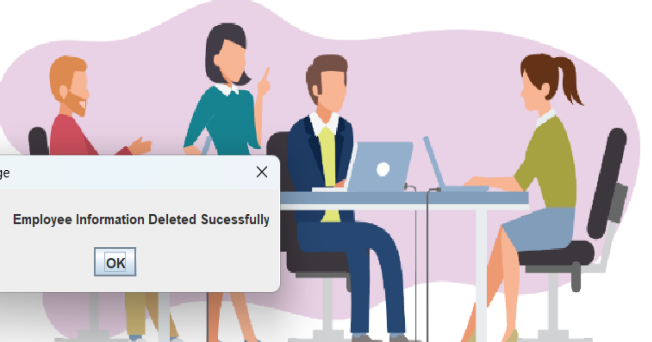
Delete

Back

Message

Employee Information Deleted Sucessfully

OK



Successful Deletion:

Search by Employee Id

122267

Search

Print

Update

Back

name	fname	dob	salary	address	phone	email	education	designation	aadhar	empld
Dhanush	Radhe Shyam	15-May-2004	5000000	def	1532649764	fhsbvhlr	MBA	Senior Analyst	8468468	122267
Madhuri	Arun	18-May-2003	5000000	ksrtbgslitdr	6846854689	difbglieue	B.COM	Buisiness An...	6816848	683941
Kalyan	Hrithik	22-Aug-2004	5000000	irubgtiebtuw	6846855225	lrtgblrstb	MCA	Data Analyst	684685165	880861

Explanation of the Source code:

Login class:

- Takes the input of the username and password

Connection class:

- Connects the database and Java compiler

View Employee class:

- Displays the details of all the employees present

Update Employee class:

- Takes the input of Employee Id, details that require changes, and updates them

Remove Employee class:

- Takes the input of Employee Id and deletes the respective employee details

Home class:

- It creates the home page which displays the operations that can be performed (Add Employee, View Employee, Update Employee, Delete Employee)

Add Employee class:

- It takes the details of an employee and adds the employee to the existing employee list

Conclusion

- This project has been a great experience for all of us as it developed our knowledge of JAVA programming and helped us improve our programming and creativity skills.
 - We used the concept of classes, inheritance, and interfaces to develop this project which is interesting as well as useful.
 - We would like to thank our guide, Sudhakar sir, for giving us the proper guidance which was much needed.
- We hope, our project met your expectations and that our efforts are reflected in this project.