

A Project Report on

EMPLOYEE MANAGEMENT SYSTEM

MASTER DEGREE DISSERTATION

**A thesis submitted in partial fulfillment of the requirements for the
degree of MSc. In Computer Science**

Submitted
by

DIPIKA RAY

Examination Roll: 15499020022

Registration No: 201541899010002 OF 2020-21

&

SANTOSH PAL

Examination Roll: 15499020016

Registration No: 201541899010008 OF 2020-21

Under the supervision of

TANUMOY NAG

**Dinabandhu Andrews Institute of Technology and Management
Maulana Abul Kalam Azad University of Technology.**

Date - June, 2022

TO WHOM AT MAY CONCERN

This is certified that the work entitled as '**EMPLOYEE MANAGEMENT SYSTEM**' has been satisfactorily completed by **DIPIKA RAY** (Registration no-201541899010002 OF 2020-21 **Roll no-15499020022**) & **SANTOSH PAL** (Registration no-201541899010008 OF 2020-21 **Roll no-15499020016**) .

It is a bonafide work carried out under my supervision at DINABANDHU ANDREWS INSTITUTE OF TECHNOLOGY AND MANAGEMENT, Kolkata for partial fulfillment of M.Sc in Computer Science during the academic year 2021-2022.

It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve for which it has been submitted.

Project Guide

TANUMOY NAG

Assistant professor

DINABANDHU ANDREWS INSTITUTE OF TECHNOLOGY AND
MANAGEMENT, Kolkata

Forward by

Paramita Ray

HOD of Computer science Dept

DINABANDHU ANDREWS INSTITUTE OF TECHNOLOGY AND
MANAGEMENT, Kolkata

Signature of the External Examiner

Date:

DECLARATION OF ORIGINALITY AND COMPLIENCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains original research work done by me, as part of Master of Computer Science studies. All information in this document has been obtained and presented in accordance with the academic rules and ethical content.

I also declare that, as required by these rules and conduct I have fully cited and referenced all the materials.

Name : DIPIKA RAY

Examination Roll: 15499020022

Registration No: 201541899010002 OF 2020-21

Name : SANTOSH PAL

Examination Roll: 15499020016

Registration No: 201541899010008 OF 2020-21

PROJECT TITLE:- EMPLOYMENT MANAGEMENT SYSTEM

SIGNATURE & DATE

ACKNOWLEDGEMENT

I would like to express my sincere thanks to all the people who have helped me most throughout my project. First of all, I am grateful to Tanumoy sir for helping us for any sort of doubt. And for explaining each and every topic of JAVA so conceptually and briefly in details. Thank you all other faculty members for helping us in building our project till end with patience.

DATE: _____

SIGNATURE: -----

DIPIKA RAY

Examination Roll: 154990200022

Registration No: 201541899010002 OF 2020-21

SIGNATURE: -----

SANTOSH PAL

Examination Roll: 154990200016

Registration No: 201541899010008 OF 2020-21

ABSTRACT

“Employee Management System” is designed to make the existing manual system automatic with the help of computerised equipment and full-edged computer software, fulfilling their requirements, so that their valuable data and information can be stored for a longer period with easy access and manipulation of the same. The required software is easily available and easy to work with. This web application can maintain and view computerised records without getting redundant entries. The project describes how to manage user data for good performance and provide better services for the client.

Table of Contents

SL No	Topics	Page No
1	Introduction	7
2	Objective	7
3	Design Approach <ul style="list-style-type: none"> • Introduction To Design • Supportive Diagrams <i>UML Diagrams , Use Case Diagrams , Sequence Diagrams, Collaboration Diagrams, Class Diagrams, State Chart Diagrams, Data Flow Diagrams, E-R Diagrams</i> 	8-11
4	Technology Used <ul style="list-style-type: none"> • Hardware • Software • Database 	12
5	Development Phases	12-13
6	Project Modules <ul style="list-style-type: none"> • < Module 1> • < Module 2> • 	13
7	Implementation <ul style="list-style-type: none"> • Functionality discussion with Screenshots • Testing 	14-
8	Conclusion / Future Work	
9	Reference / Bibliography	

INTRODUCTION

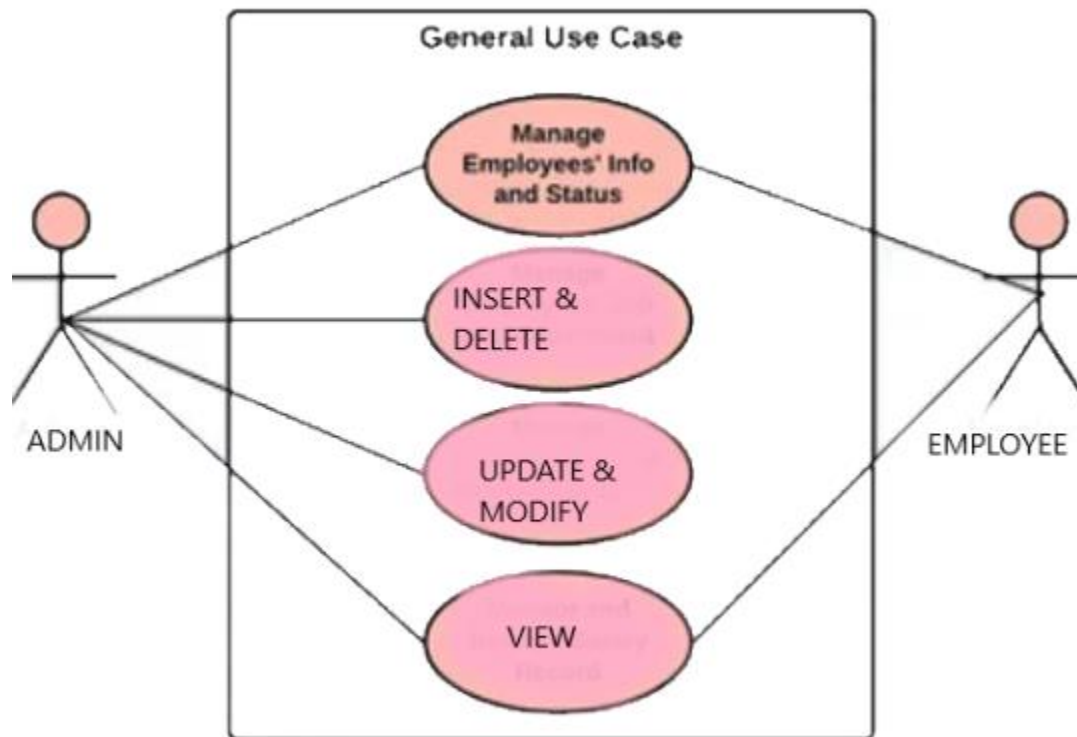
The proposed project “Employee Management System” has been developed to overcome the problems faced in the practicing of manual system. This software is built to eliminate and in some cases reduce the hardships faced by the existing system. Moreover this system is designed for particular need of the company to carry out its operations in a smooth and effective manner. This web application is reduced as much as possible to avoid errors while entering data. It also provides error message while entering invalid data. It is user-friendly as no formal knowledge is required to use the system. Human resource challenges are faced by every organization which has to be overcome by the organization. Every organization has different employee and payroll management needs. Therefore I have design exclusive Employee and payroll Management System that are adapted to the organization’s Managerial Requirements.

OBJECTIVES

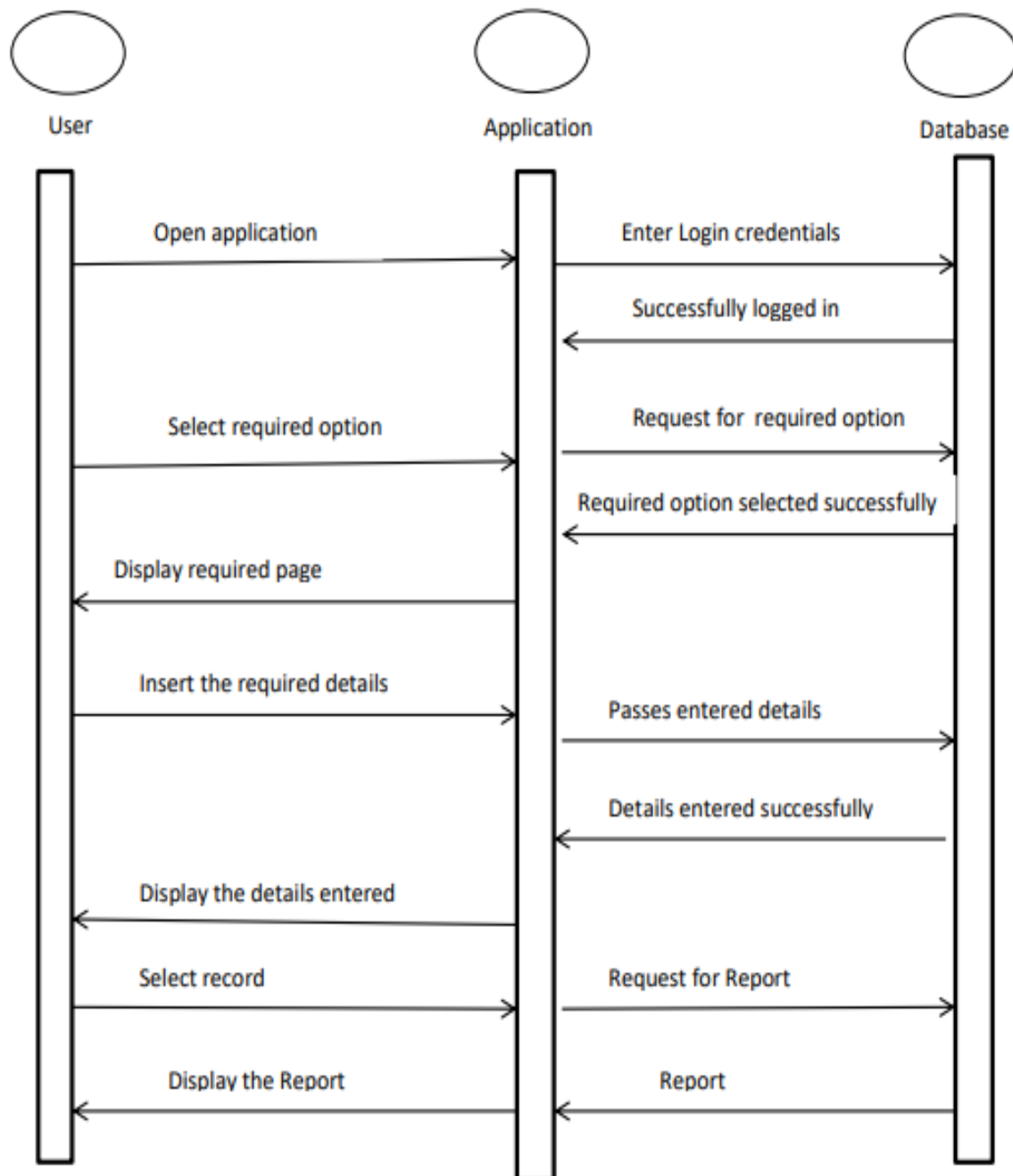
The purpose of this document is to describe the functionality and specifications of the design of a web application for Managing Employees and their payroll. The expected audiences of this document are the developers and the admin of the web application. Now with the help of this system the admin has the information on his finger tips and can easily prepare a good record based on their requirements. Finally, we can say that this system will not only automate the process but save the valuable time of the manager or the admin, which can be well utilized buy his institute. This will be an additional advantage and management of power based on their free time from his normal duty.

DESIGN APPROACH

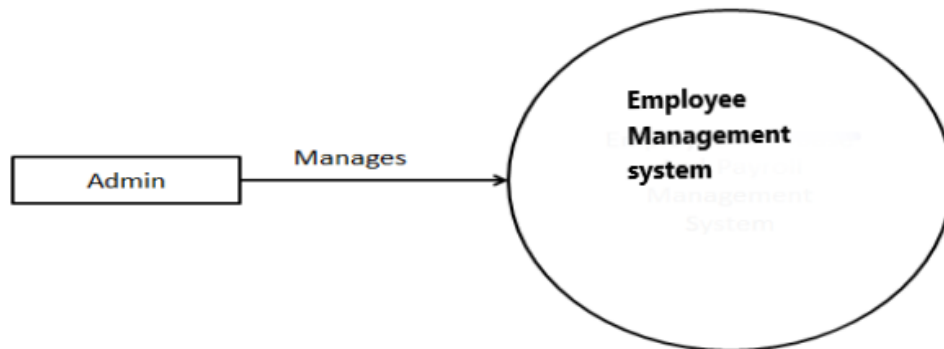
Use Case Diagram:



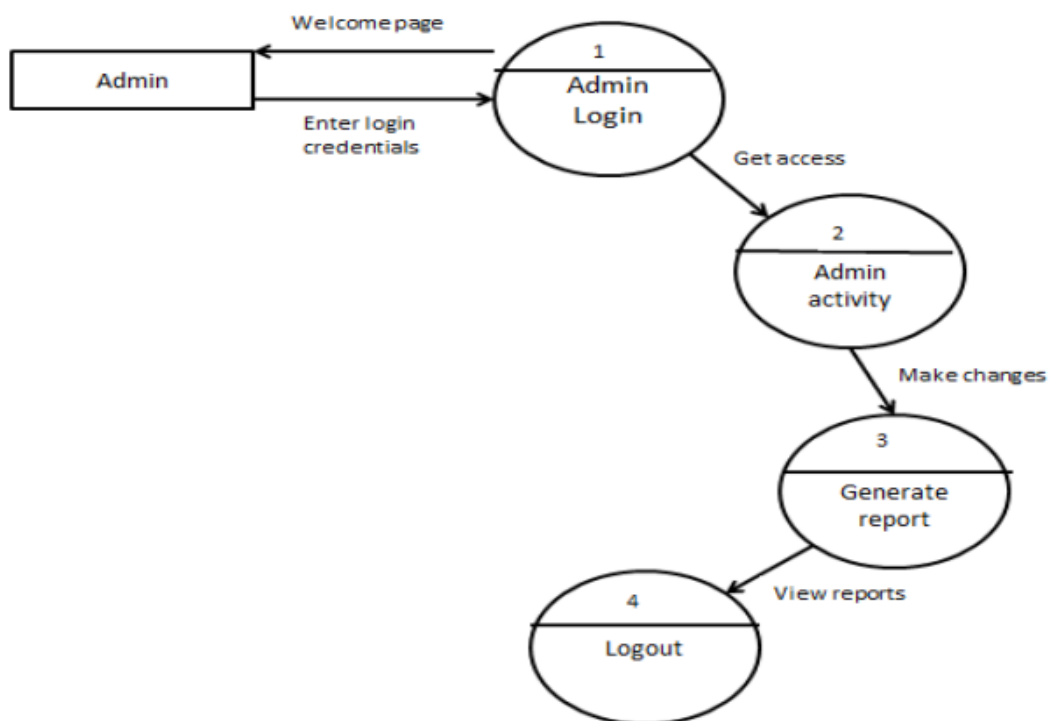
Sequence Diagram:



DFD(Data Flow Diagram)



Level 0



Level 1

Entity Relationship Diagram(ERD)

TECHNOLOGY USED:

1. Hardware:

Device : Laptop

Processor : 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz

RAM : 8.00 GB

System Type: 64-bit operating system, x64-based processor

2. Software:

- i) MYSQL workbench 8.0 CE
- ii) Apache Netbeans 13
- iii) mysql-connector-java-8.0.29
- iv) jdk-17.0.1

3. Database:

- i) MYSQL

DEVELOPMENT PHASE:

After identifying the scope of the project, the feasibility study is needed to be carried out. It is basically keeping the following points in mind.

Building the software for meeting the scope: This software has met the scope. As there is no data involved in the system, processing on the file, and the behaviour of this project is already identified and bundled in quantitative manner. The processing of this software is very simple as it has been designed in java and it has been well divided into several functions according to the need.

Technically feasible: This software is very much technically feasible. This software is very much concerned with specifying equipment and the software will successfully satisfy almost all

the admin's requirements. The technical need for this system may vary considerably but might include:

- a. The facility to produce output in a given time.
- b. Response time under certain conditions.
- c. Ability to process data at a particular speed.
























Therefore, the basic input/output of data is identified. So, the project can easily be build up and it will also be technically feasible.

State of Art: The project is very much within the state of art since the project is a WINDOWS based; it uses very modern and common technique. Beside it is very much modern and user friendly. It also works as middleware i.e. only in between the user and the file. So, it is completely a state of art project.

Financially Feasible: The project is very much financially feasible. The implementation and development cost of this software under the reach of any college. Moreover, it requires some training for the use. So, training cost can be neglected and the resources of this software are very much available. It also reduces the labour and extra cost to be paid for labour. So indeed, it is financially feasible.

Resources: As motioned earlier that the resources are easily available and the cost of training is almost negligible. Sometimes situations may arise when it may not be so much easy. For a person completely unaware of using a computer system could result in a training cost or for a very small organization the purchase of a computer, instalment of the system and other charges may lead to a difficult matter.

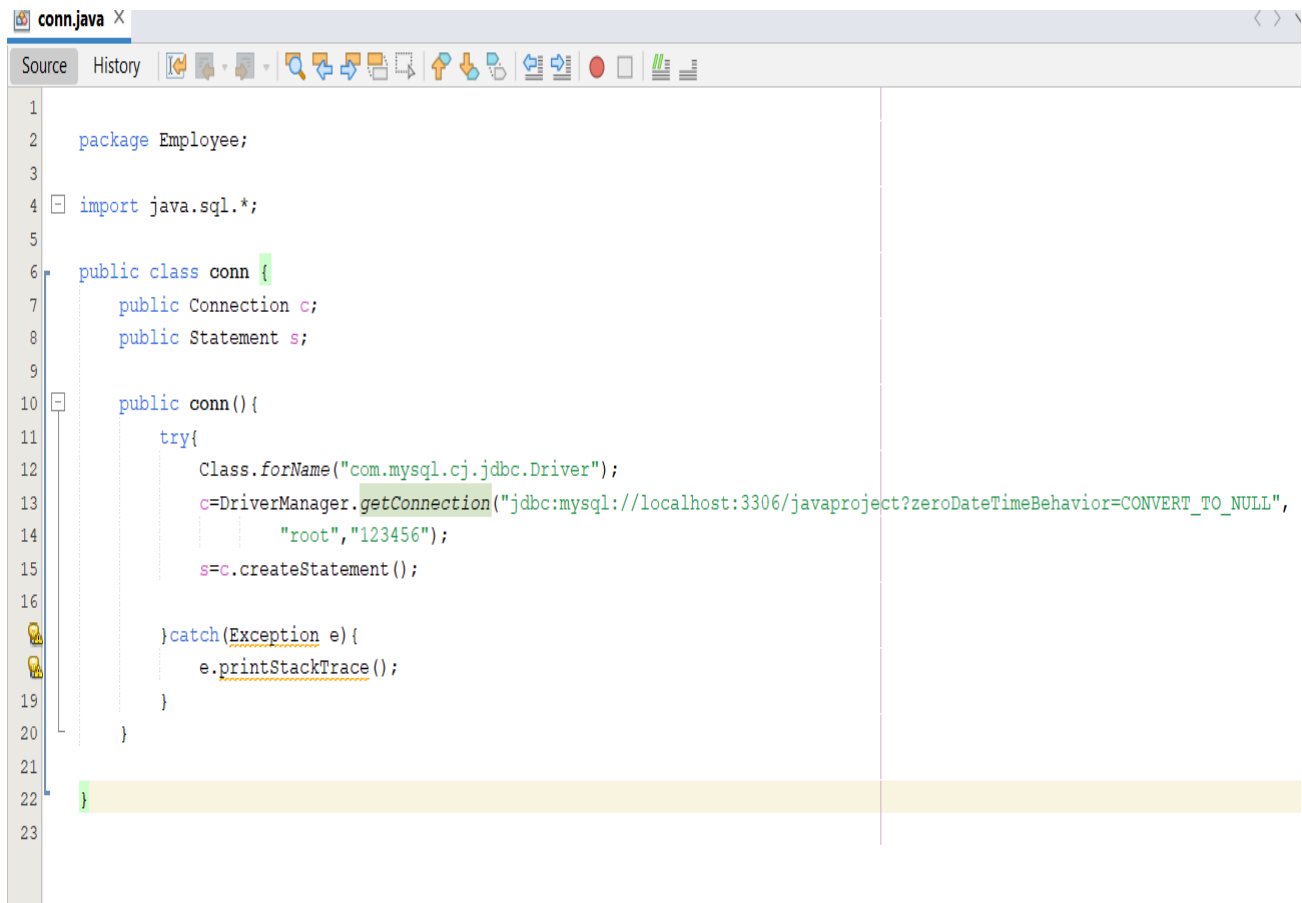
PROJECT MODULES:

- ▼  Employee Management System
 - ▼  Source Packages
 - ▼  Employee
 -  Add_Employee.java
 -  Print_Data.java
 -  Remove_Employee.java
 -  Search_Employee.java
 -  Update_Employee.java
 -  View_Employee.java
 -  conn.java
 -  details.java
 -  login.java
 - ▼  icon
 -  add_emp3.jpg
 -  details.jpg
 -  login.jpg
 -  print.jpg
 -  remove.jpg
 -  view.jpg
 - >  Test Packages
 - ▼  Libraries
 - >  mysql-connector-java-8.0.29.jar
 - >  JDK 17 (Default)

IMPLEMENTATION

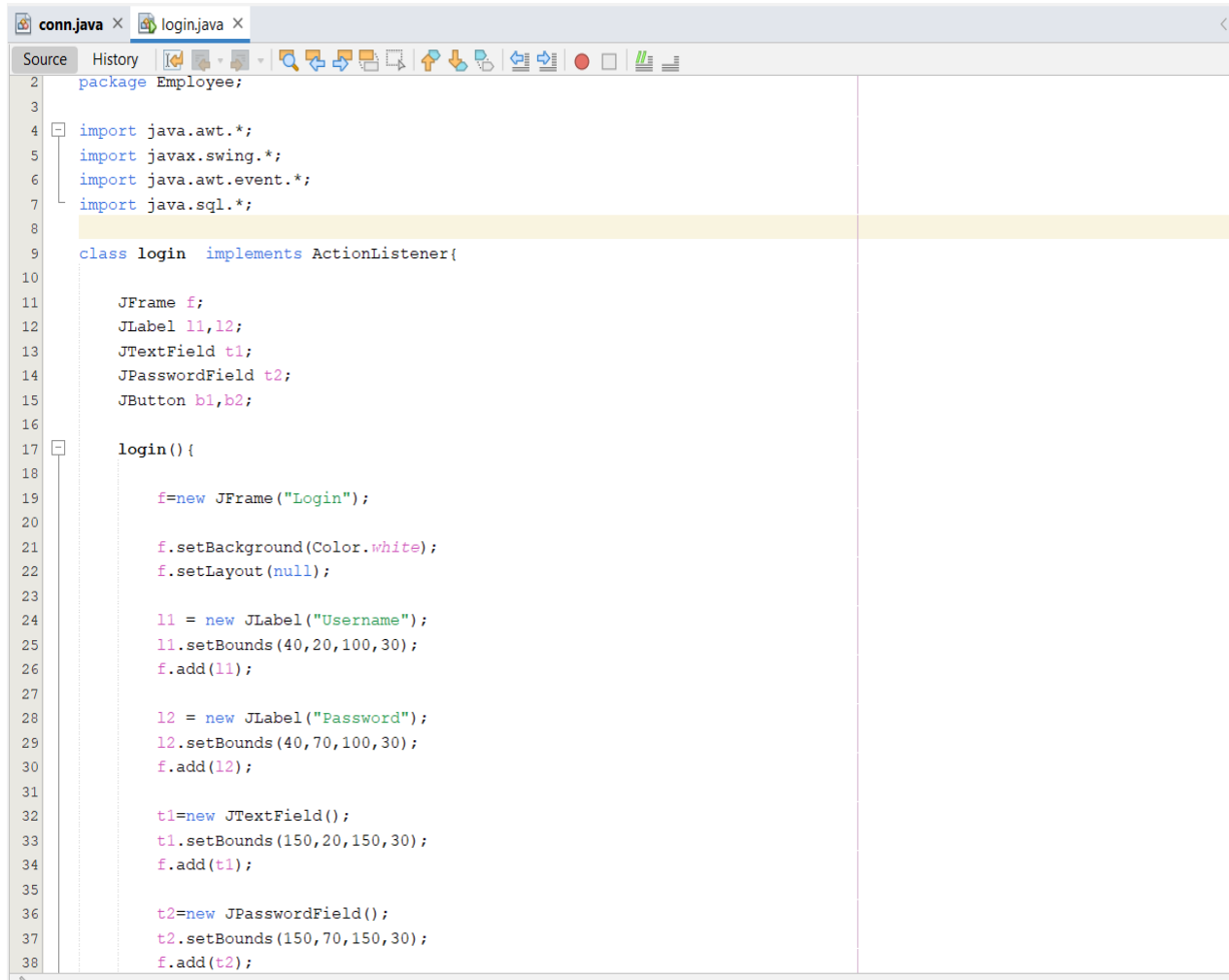
1)SOURCE CODE:

Class: [conn.java]

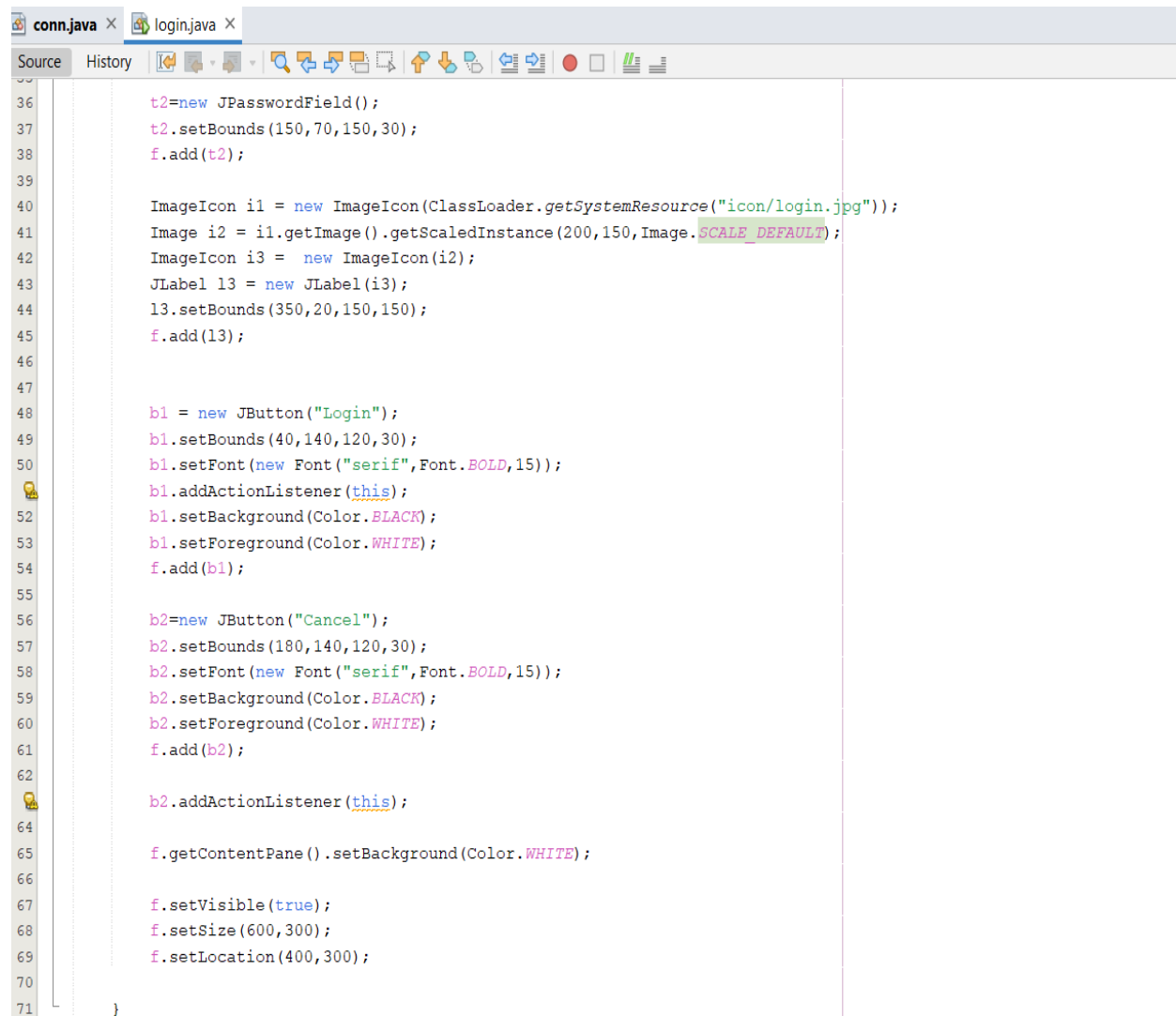


```
1
2 package Employee;
3
4 import java.sql.*;
5
6 public class conn {
7     public Connection c;
8     public Statement s;
9
10    public conn(){
11        try{
12            Class.forName("com.mysql.cj.jdbc.Driver");
13            c=DriverManager.getConnection("jdbc:mysql://localhost:3306/javaproject?zeroDateTimeBehavior=CONVERT_TO_NULL",
14                "root","123456");
15            s=c.createStatement();
16
17        }catch(Exception e){
18            e.printStackTrace();
19        }
20    }
21
22 }
23
```

Class:[login.java]

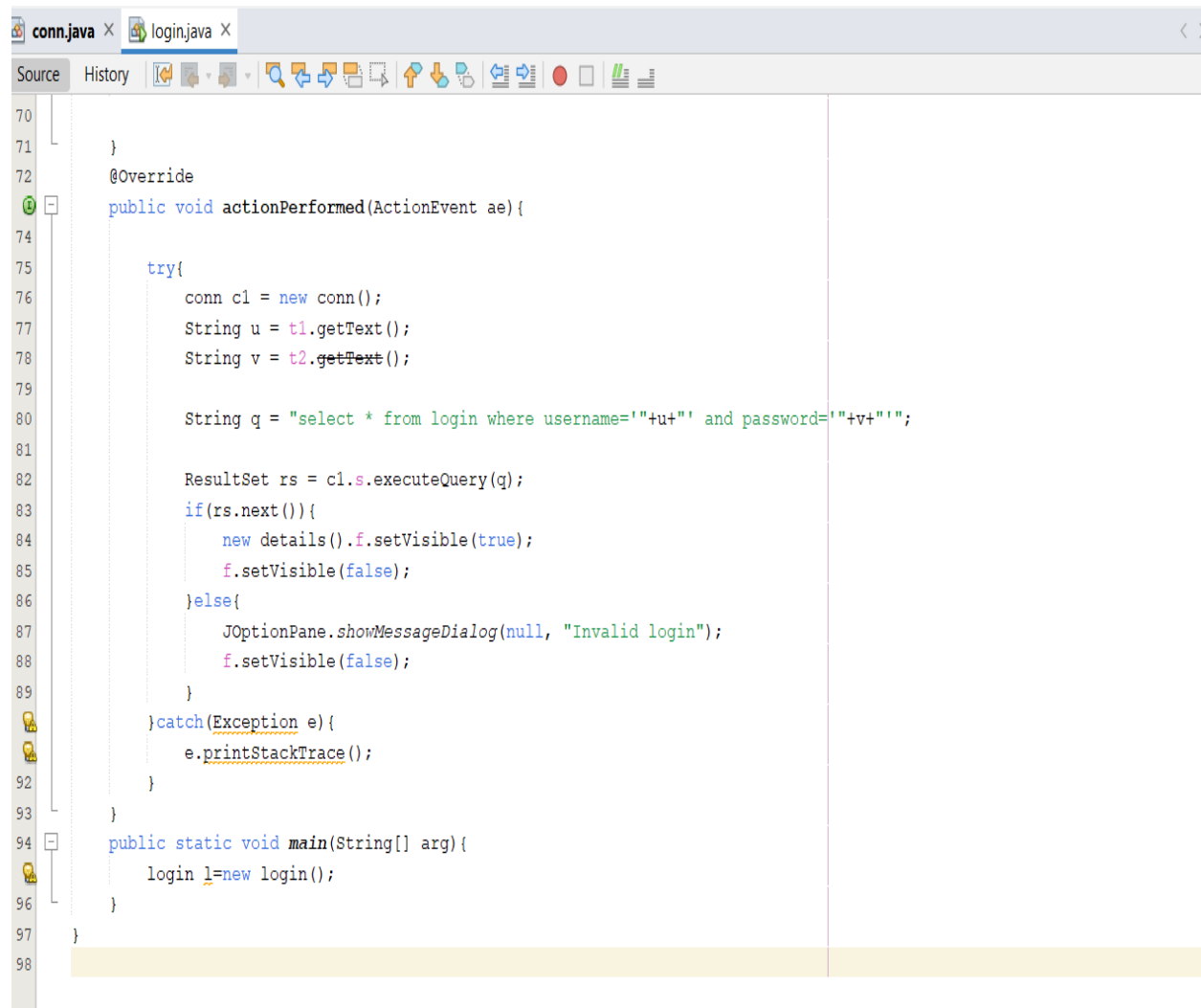


```
2 package Employee;
3
4 import java.awt.*;
5 import javax.swing.*;
6 import java.awt.event.*;
7 import java.sql.*;
8
9 class login implements ActionListener{
10
11     JFrame f;
12     JLabel l1,l2;
13     JTextField t1;
14     JPasswordField t2;
15     JButton b1,b2;
16
17     login() {
18
19         f=new JFrame("Login");
20
21         f.setBackground(Color.white);
22         f.setLayout(null);
23
24         l1 = new JLabel("Username");
25         l1.setBounds(40,20,100,30);
26         f.add(l1);
27
28         l2 = new JLabel("Password");
29         l2.setBounds(40,70,100,30);
30         f.add(l2);
31
32         t1=new JTextField();
33         t1.setBounds(150,20,150,30);
34         f.add(t1);
35
36         t2=new JPasswordField();
37         t2.setBounds(150,70,150,30);
38         f.add(t2);
```

The screenshot shows an IDE with two tabs: 'conn.java' and 'login.java'. The 'login.java' tab is active, displaying Java code for a login window. The code includes imports for 'JPasswordField', 'ImageIcon', 'Image', 'JLabel', 'JButton', 'Font', and 'Color'. The main logic is enclosed in a 'try' block. It creates a password field 't2', a login button 'b1', and a cancel button 'b2'. The login button is styled with a black background and white text. The window's content pane is set to white. The window is then made visible, sized to 600x300, and positioned at (400, 300). The code ends with a closing brace for the 'try' block.

```
36     t2=new JPasswordField();
37     t2.setBounds(150,70,150,30);
38     f.add(t2);
39
40     ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icon/login.jpg"));
41     Image i2 = i1.getImage().getScaledInstance(200,150,Image.SCALE_DEFAULT);
42     ImageIcon i3 = new ImageIcon(i2);
43     JLabel l3 = new JLabel(i3);
44     l3.setBounds(350,20,150,150);
45     f.add(l3);
46
47
48     b1 = new JButton("Login");
49     b1.setBounds(40,140,120,30);
50     b1.setFont(new Font("serif",Font.BOLD,15));
51     b1.addActionListener(this);
52     b1.setBackground(Color.BLACK);
53     b1.setForeground(Color.WHITE);
54     f.add(b1);
55
56     b2=new JButton("Cancel");
57     b2.setBounds(180,140,120,30);
58     b2.setFont(new Font("serif",Font.BOLD,15));
59     b2.setBackground(Color.BLACK);
60     b2.setForeground(Color.WHITE);
61     f.add(b2);
62
63     b2.addActionListener(this);
64
65     f.getContentPane().setBackground(Color.WHITE);
66
67     f.setVisible(true);
68     f.setSize(600,300);
69     f.setLocation(400,300);
70
71 }
```



```
70  
71  
72     @Override  
73     public void actionPerformed(ActionEvent ae){  
74  
75         try{  
76             conn c1 = new conn();  
77             String u = t1.getText();  
78             String v = t2.getText();  
79  
80             String q = "select * from login where username='"+u+"' and password='"+v+"'";  
81  
82             ResultSet rs = c1.s.executeQuery(q);  
83             if(rs.next()){  
84                 new details().f.setVisible(true);  
85                 f.setVisible(false);  
86             }else{  
87                 JOptionPane.showMessageDialog(null, "Invalid login");  
88                 f.setVisible(false);  
89             }  
90         }catch(Exception e){  
91             e.printStackTrace();  
92         }  
93     }  
94     public static void main(String[] arg){  
95         login l=new login();  
96     }  
97 }  
98
```

Output:

 Login—□✕

Username

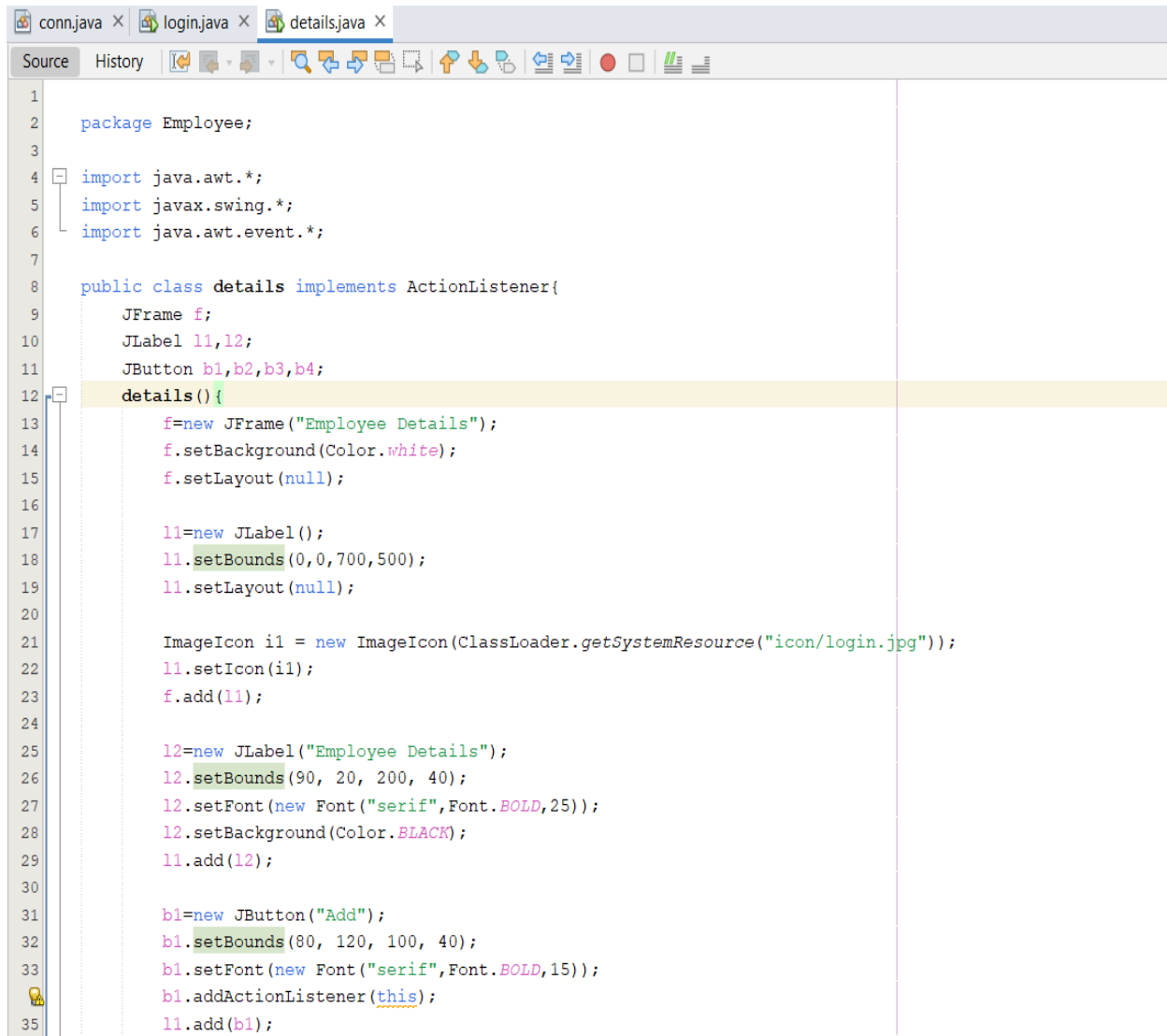
Password

Login

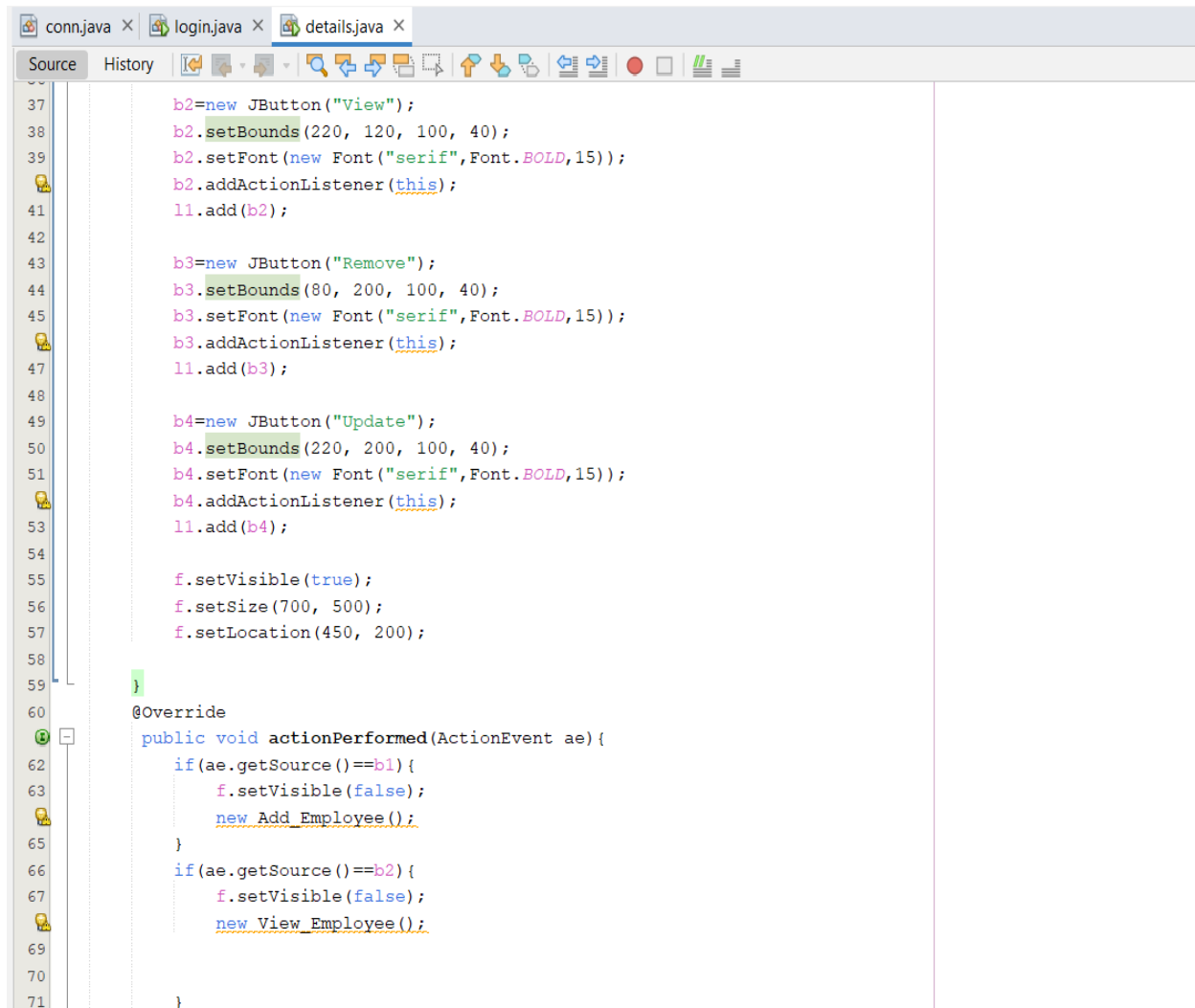
Cancel



Class:[details.java]



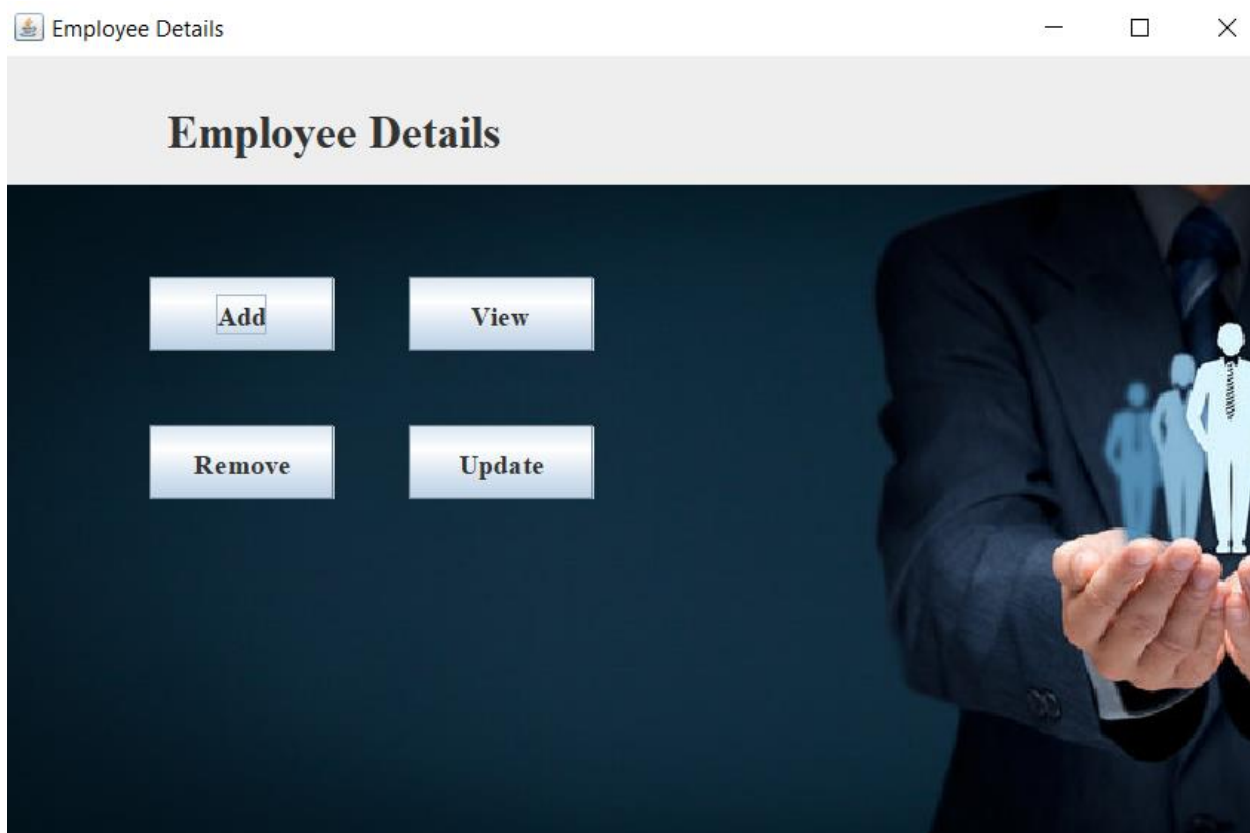
```
1
2 package Employee;
3
4 import java.awt.*;
5 import javax.swing.*;
6 import java.awt.event.*;
7
8 public class details implements ActionListener{
9     JFrame f;
10    JLabel l1,l2;
11    JButton b1,b2,b3,b4;
12    details(){
13        f=new JFrame("Employee Details");
14        f.setBackground(Color.white);
15        f.setLayout(null);
16
17        l1=new JLabel();
18        l1.setBounds(0,0,700,500);
19        l1.setLayout(null);
20
21        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icon/login.jpg"));
22        l1.setIcon(i1);
23        f.add(l1);
24
25        l2=new JLabel("Employee Details");
26        l2.setBounds(90, 20, 200, 40);
27        l2.setFont(new Font("serif",Font.BOLD,25));
28        l2.setBackground(Color.BLACK);
29        l1.add(l2);
30
31        b1=new JButton("Add");
32        b1.setBounds(80, 120, 100, 40);
33        b1.setFont(new Font("serif",Font.BOLD,15));
34        b1.addActionListener(this);
35        l1.add(b1);
```



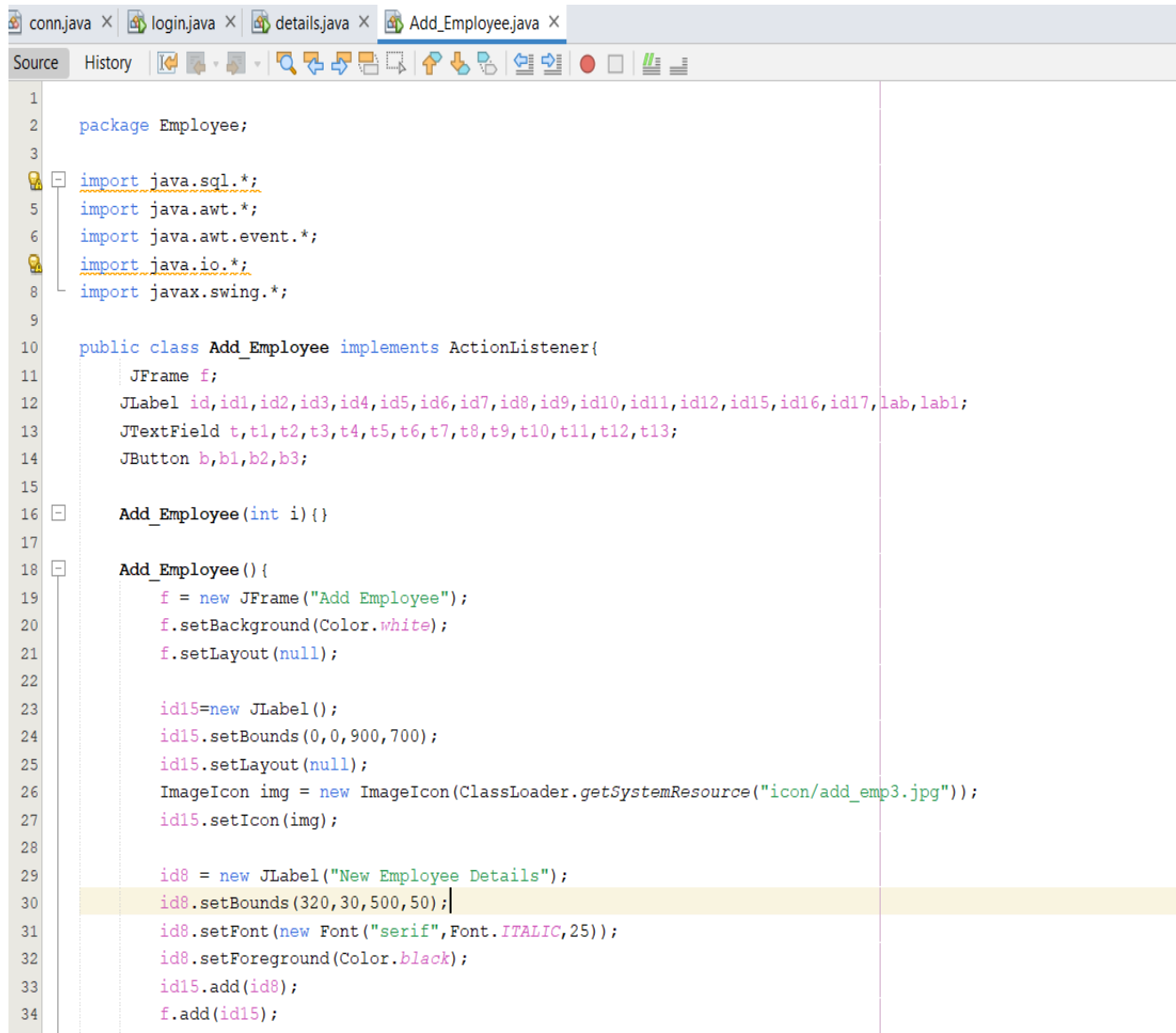
```
37     b2=new JButton("View");
38     b2.setBounds(220, 120, 100, 40);
39     b2.setFont(new Font("serif",Font.BOLD,15));
40     b2.addActionListener(this);
41     l1.add(b2);
42
43     b3=new JButton("Remove");
44     b3.setBounds(80, 200, 100, 40);
45     b3.setFont(new Font("serif",Font.BOLD,15));
46     b3.addActionListener(this);
47     l1.add(b3);
48
49     b4=new JButton("Update");
50     b4.setBounds(220, 200, 100, 40);
51     b4.setFont(new Font("serif",Font.BOLD,15));
52     b4.addActionListener(this);
53     l1.add(b4);
54
55     f.setVisible(true);
56     f.setSize(700, 500);
57     f.setLocation(450, 200);
58
59 }
60 @Override
61 public void actionPerformed(ActionEvent ae){
62     if(ae.getSource()==b1){
63         f.setVisible(false);
64         new Add_Employee();
65     }
66     if(ae.getSource()==b2){
67         f.setVisible(false);
68         new View_Employee();
69     }
70
71 }
```

```
conn.java x login.java x details.java x
Source History
63         f.setVisible(false);
64         new Add_Employee();
65     }
66     if(ae.getSource()==b2){
67         f.setVisible(false);
68         new View_Employee();
69     }
70
71     }
72     if(ae.getSource()==b3){
73         f.setVisible(false);
74         new Remove_Employee();
75     }
76     if(ae.getSource()==b4){
77         f.setVisible(false);
78         new Search_Employee();
79     }
80 }
81
82
83 public static void main(String[] arg){
84     details d=new details();
85 }
86 }
87
```

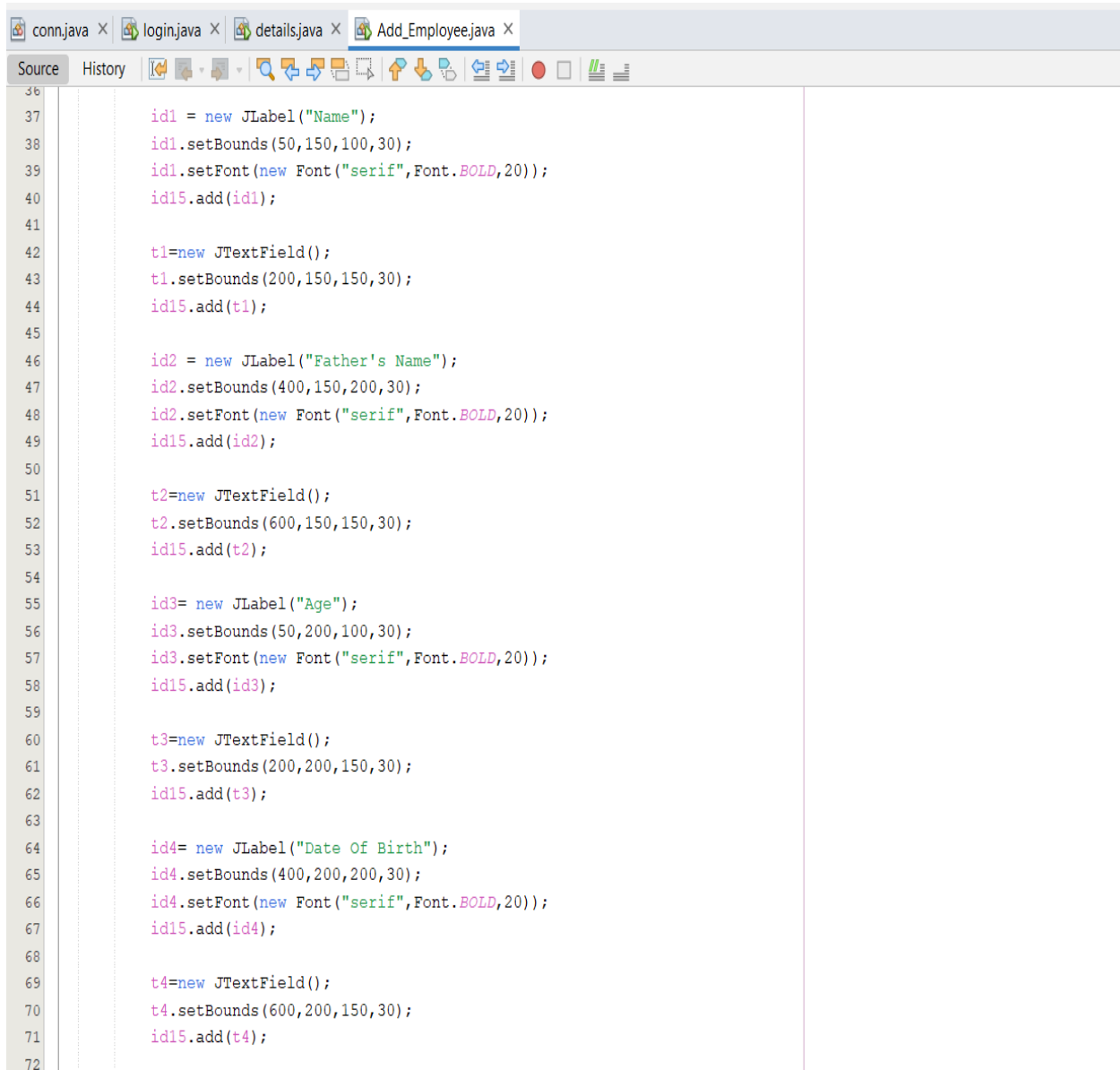
Output:



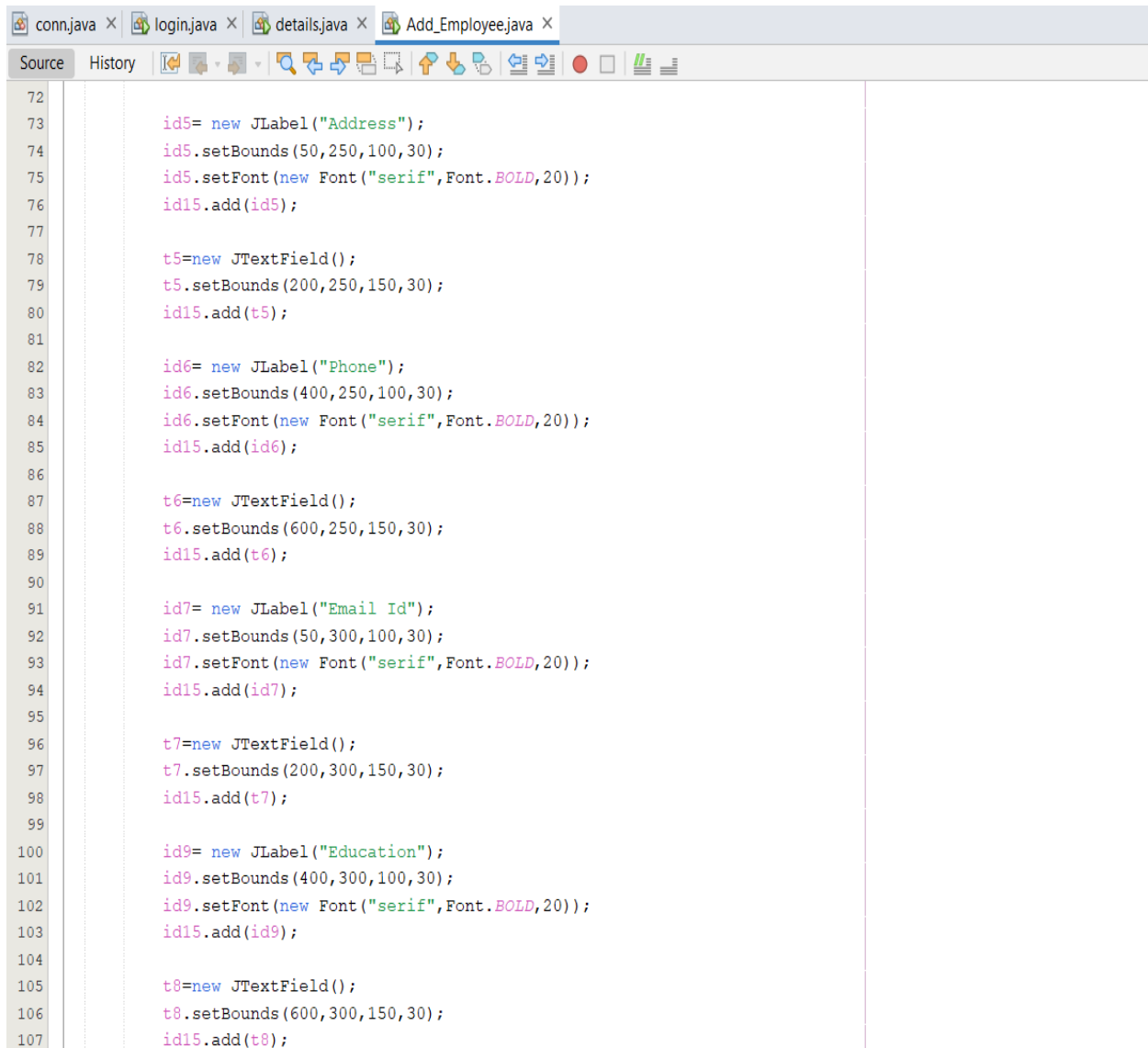
Class:[Add_Employee.java]



```
1 package Employee;
2
3
4 import java.sql.*;
5 import java.awt.*;
6 import java.awt.event.*;
7 import java.io.*;
8 import javax.swing.*;
9
10 public class Add_Employee implements ActionListener{
11     JFrame f;
12     JLabel id,id1,id2,id3,id4,id5,id6,id7,id8,id9,id10,id11,id12,id15,id16,id17,lab,lab1;
13     JTextField t,t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13;
14     JButton b,b1,b2,b3;
15
16     Add_Employee(int i){}
17
18     Add_Employee(){
19         f = new JFrame("Add Employee");
20         f.setBackground(Color.white);
21         f.setLayout(null);
22
23         id15=new JLabel();
24         id15.setBounds(0,0,900,700);
25         id15.setLayout(null);
26         ImageIcon img = new ImageIcon(ClassLoader.getResource("icon/add_emp3.jpg"));
27         id15.setIcon(img);
28
29         id8 = new JLabel("New Employee Details");
30         id8.setBounds(320,30,500,50);
31         id8.setFont(new Font("serif",Font.ITALIC,25));
32         id8.setForeground(Color.black);
33         id15.add(id8);
34         f.add(id15);
```



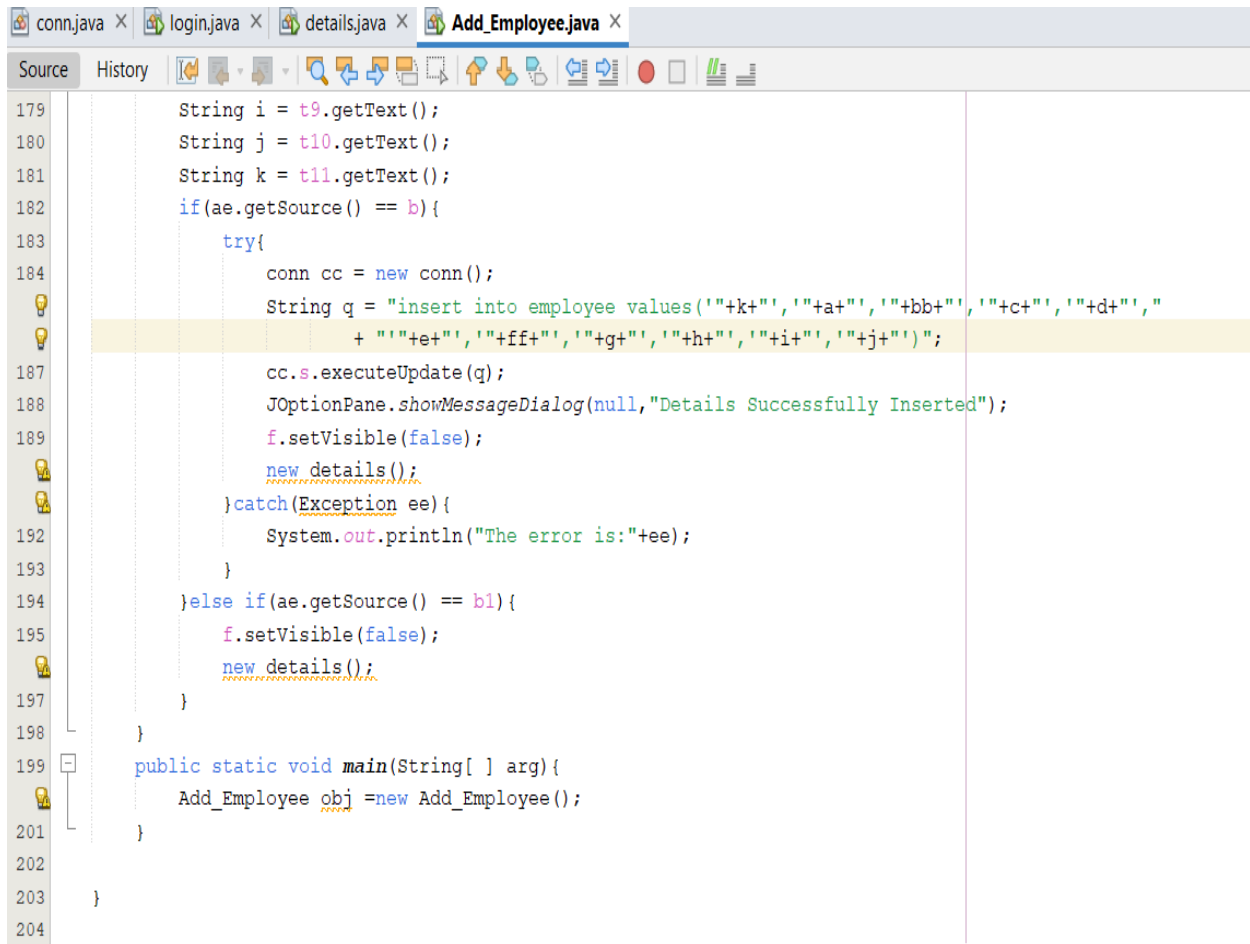
```
36
37     id1 = new JLabel("Name");
38     id1.setBounds(50,150,100,30);
39     id1.setFont(new Font("serif",Font.BOLD,20));
40     id15.add(id1);
41
42     t1=new JTextField();
43     t1.setBounds(200,150,150,30);
44     id15.add(t1);
45
46     id2 = new JLabel("Father's Name");
47     id2.setBounds(400,150,200,30);
48     id2.setFont(new Font("serif",Font.BOLD,20));
49     id15.add(id2);
50
51     t2=new JTextField();
52     t2.setBounds(600,150,150,30);
53     id15.add(t2);
54
55     id3= new JLabel("Age");
56     id3.setBounds(50,200,100,30);
57     id3.setFont(new Font("serif",Font.BOLD,20));
58     id15.add(id3);
59
60     t3=new JTextField();
61     t3.setBounds(200,200,150,30);
62     id15.add(t3);
63
64     id4= new JLabel("Date Of Birth");
65     id4.setBounds(400,200,200,30);
66     id4.setFont(new Font("serif",Font.BOLD,20));
67     id15.add(id4);
68
69     t4=new JTextField();
70     t4.setBounds(600,200,150,30);
71     id15.add(t4);
72
```

```
72
73     id5= new JLabel("Address");
74     id5.setBounds(50,250,100,30);
75     id5.setFont(new Font("serif",Font.BOLD,20));
76     id15.add(id5);
77
78     t5=new JTextField();
79     t5.setBounds(200,250,150,30);
80     id15.add(t5);
81
82     id6= new JLabel("Phone");
83     id6.setBounds(400,250,100,30);
84     id6.setFont(new Font("serif",Font.BOLD,20));
85     id15.add(id6);
86
87     t6=new JTextField();
88     t6.setBounds(600,250,150,30);
89     id15.add(t6);
90
91     id7= new JLabel("Email Id");
92     id7.setBounds(50,300,100,30);
93     id7.setFont(new Font("serif",Font.BOLD,20));
94     id15.add(id7);
95
96     t7=new JTextField();
97     t7.setBounds(200,300,150,30);
98     id15.add(t7);
99
100    id9= new JLabel("Education");
101    id9.setBounds(400,300,100,30);
102    id9.setFont(new Font("serif",Font.BOLD,20));
103    id15.add(id9);
104
105    t8=new JTextField();
106    t8.setBounds(600,300,150,30);
107    id15.add(t8);
```


```
conn.java x login.java x details.java x Add_Employee.java x
Source History
109     id10= new JLabel("Job Post");
110     id10.setBounds(50,350,100,30);
111     id10.setFont(new Font("serif",Font.BOLD,20));
112     id15.add(id10);
113
114     t9=new JTextField();
115     t9.setBounds(200,350,150,30);
116     id15.add(t9);
117
118
119     id11= new JLabel("Aadhar No");
120     id11.setBounds(400,350,100,30);
121     id11.setFont(new Font("serif",Font.BOLD,20));
122     id15.add(id11);
123
124     t10=new JTextField();
125     t10.setBounds(600,350,150,30);
126     id15.add(t10);
127
128     id12= new JLabel("Employee Id");
129     id12.setBounds(50,400,150,30);
130     id12.setFont(new Font("serif",Font.BOLD,20));
131     id15.add(id12);
132
133     t11=new JTextField();
134     t11.setBounds(200,400,150,30);
135     id15.add(t11);
136
137
138
139     lab=new JLabel();
140     lab.setBounds(200,450,250,200);
141     id15.add(lab);
142
143     lab1=new JLabel("");
144     lab1.setBounds(600,450,250,200);
145     id15.add(lab1);
```

```
conn.java x login.java x details.java x Add_Employee.java x
Source History
147     b = new JButton("Submit");
148     b.setBackground(Color.BLACK);
149     b.setForeground(Color.WHITE);
150     b.setBounds(250,550,150,40);
151
152     id15.add(b);
153
154     b1=new JButton("Cancel");
155     b1.setBackground(Color.BLACK);
156     b1.setForeground(Color.WHITE);
157     b1.setBounds(450,550,150,40);
158
159     id15.add(b1);
160
161     b.addActionListener(this);
162     b1.addActionListener(this);
163
164     f.setVisible(true);
165     f.setSize(900,700);
166     f.setLocation(400,150);
167 }
168 @Override
169 public void actionPerformed(ActionEvent ae){
170
171     String a = t1.getText();
172     String bb = t2.getText();
173     String c = t3.getText();
174     String d = t4.getText();
175     String e = t5.getText();
176     String ff = t6.getText();
177     String g = t7.getText();
178     String h = t8.getText();
179     String i = t9.getText();
180     String j = t10.getText();
181     String k = t11.getText();
```



```
179 String i = t9.getText();
180 String j = t10.getText();
181 String k = t11.getText();
182 if(ae.getSource() == b){
183     try{
184         conn cc = new conn();
185         String q = "insert into employee values('"+k+"','"+a+"','"+bb+"','"+c+"','"+d+"','"+
186             + "'"+e+"','"+ff+"','"+g+"','"+h+"','"+i+"','"+j+"')";
187         cc.s.executeUpdate(q);
188         JOptionPane.showMessageDialog(null,"Details Successfully Inserted");
189         f.setVisible(false);
190         new details();
191     }catch(Exception ee){
192         System.out.println("The error is:"+ee);
193     }
194 }else if(ae.getSource() == b1){
195     f.setVisible(false);
196     new details();
197 }
198 }
199 public static void main(String[] arg){
200     Add_Employee obj =new Add_Employee();
201 }
202
203 }
204
```

Output:

 Add Employee

— □ ×

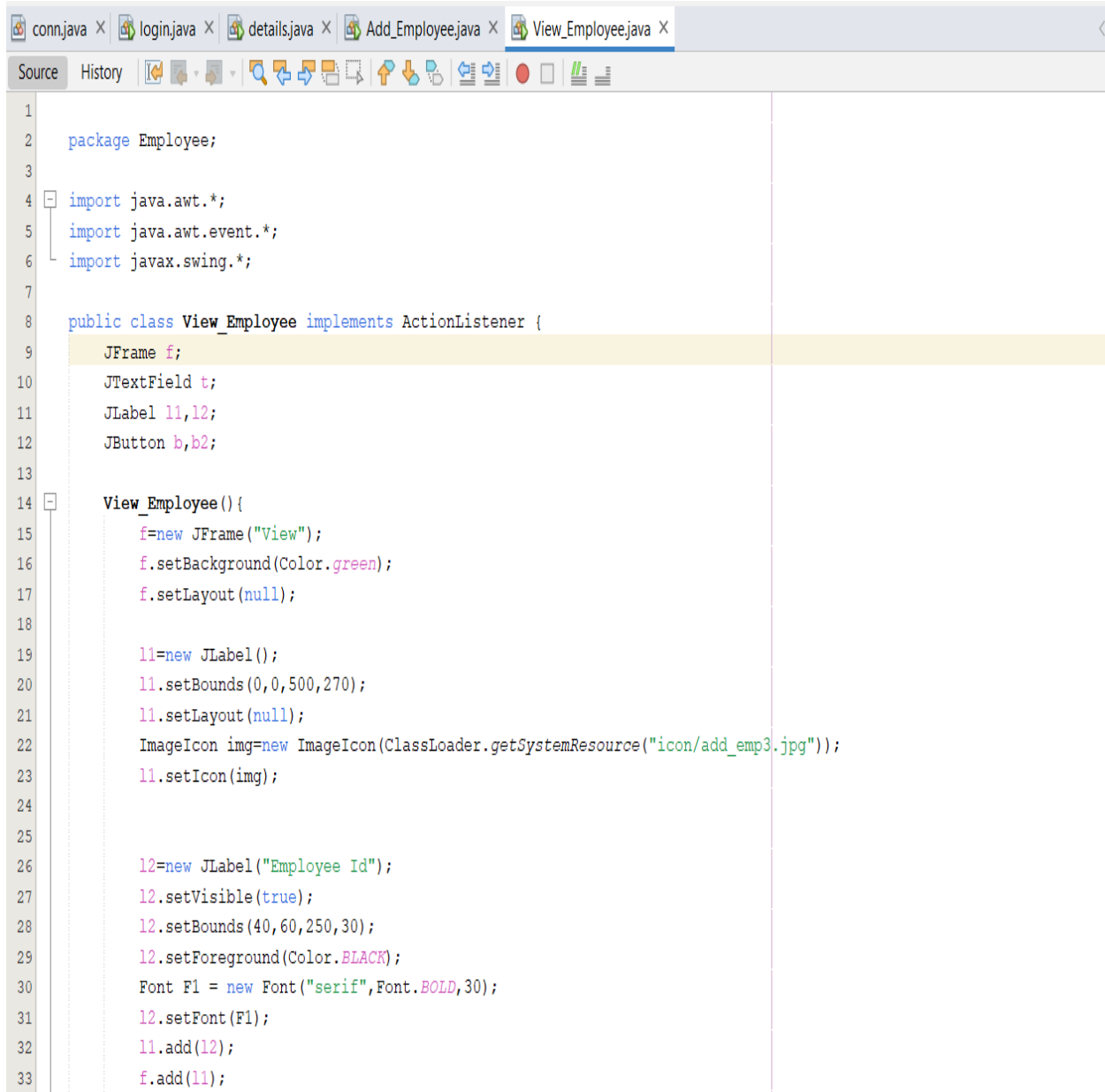
New Employee Details

Name	<input type="text"/>	Father's Name	<input type="text"/>
Age	<input type="text"/>	Date Of Birth	<input type="text"/>
Address	<input type="text"/>	Phone	<input type="text"/>
Email Id	<input type="text"/>	Education	<input type="text"/>
Job Post	<input type="text"/>	Aadhar No	<input type="text"/>
Employee Id	<input type="text"/>		

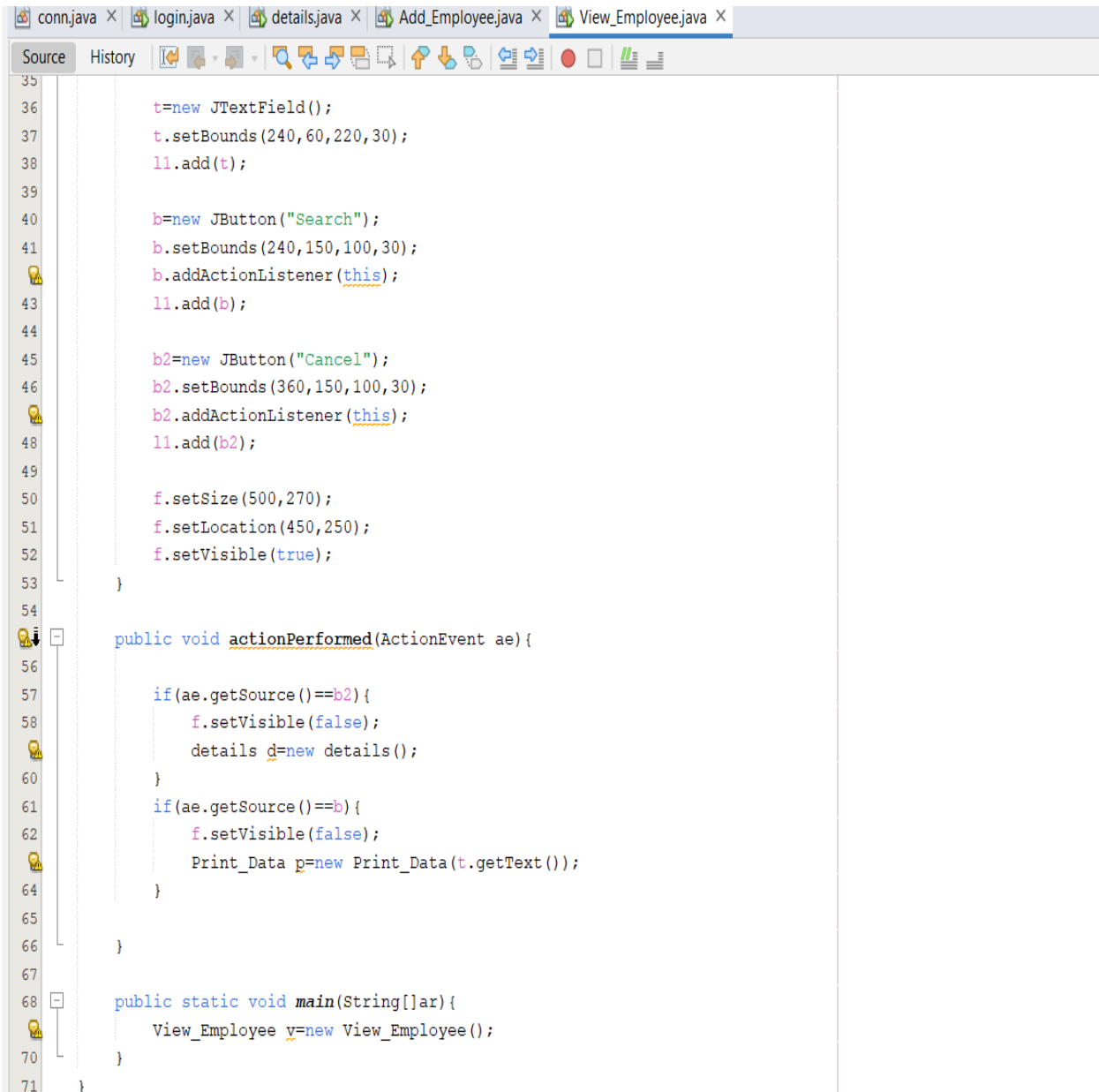
Submit

Cancel

Class:[View_Employee.java]

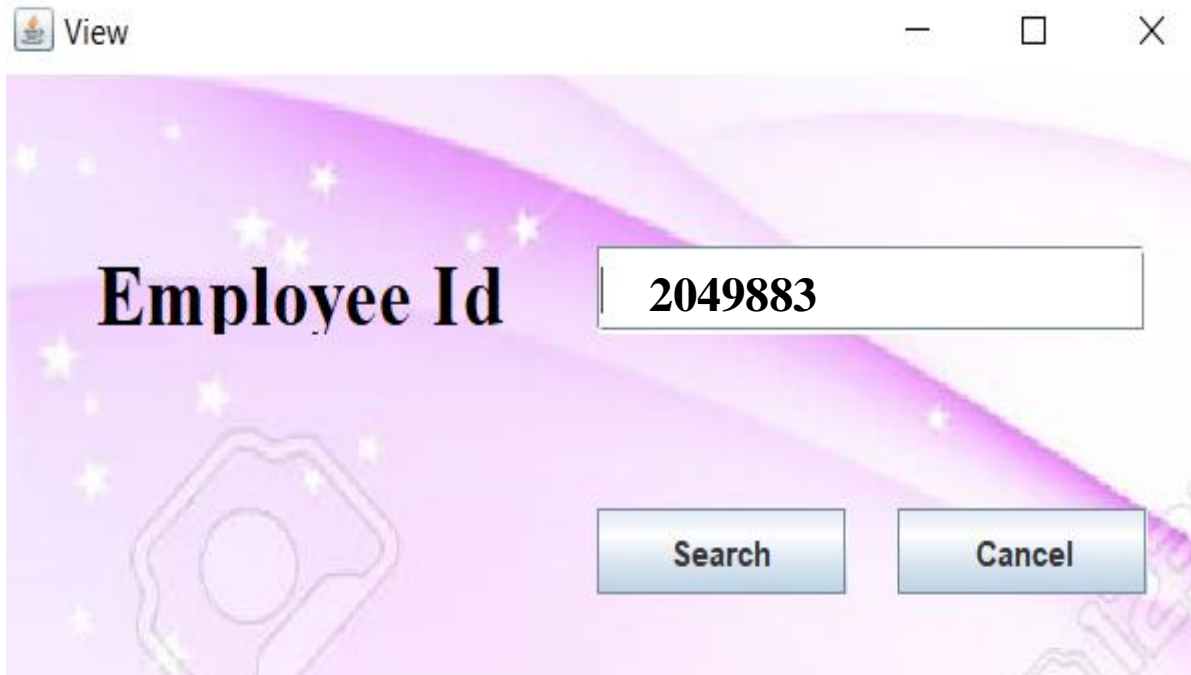


```
1
2 package Employee;
3
4 import java.awt.*;
5 import java.awt.event.*;
6 import javax.swing.*;
7
8 public class View_Employee implements ActionListener {
9     JFrame f;
10    JTextField t;
11    JLabel l1,l2;
12    JButton b,b2;
13
14    View_Employee() {
15        f=new JFrame("View");
16        f.setBackground(Color.green);
17        f.setLayout(null);
18
19        l1=new JLabel();
20        l1.setBounds(0,0,500,270);
21        l1.setLayout(null);
22        ImageIcon img=new ImageIcon(ClassLoader.getResource("icon/add_emp3.jpg"));
23        l1.setIcon(img);
24
25
26        l2=new JLabel("Employee Id");
27        l2.setVisible(true);
28        l2.setBounds(40,60,250,30);
29        l2.setForeground(Color.BLACK);
30        Font f1 = new Font("serif",Font.BOLD,30);
31        l2.setFont(f1);
32        l1.add(l2);
33        f.add(l1);
```



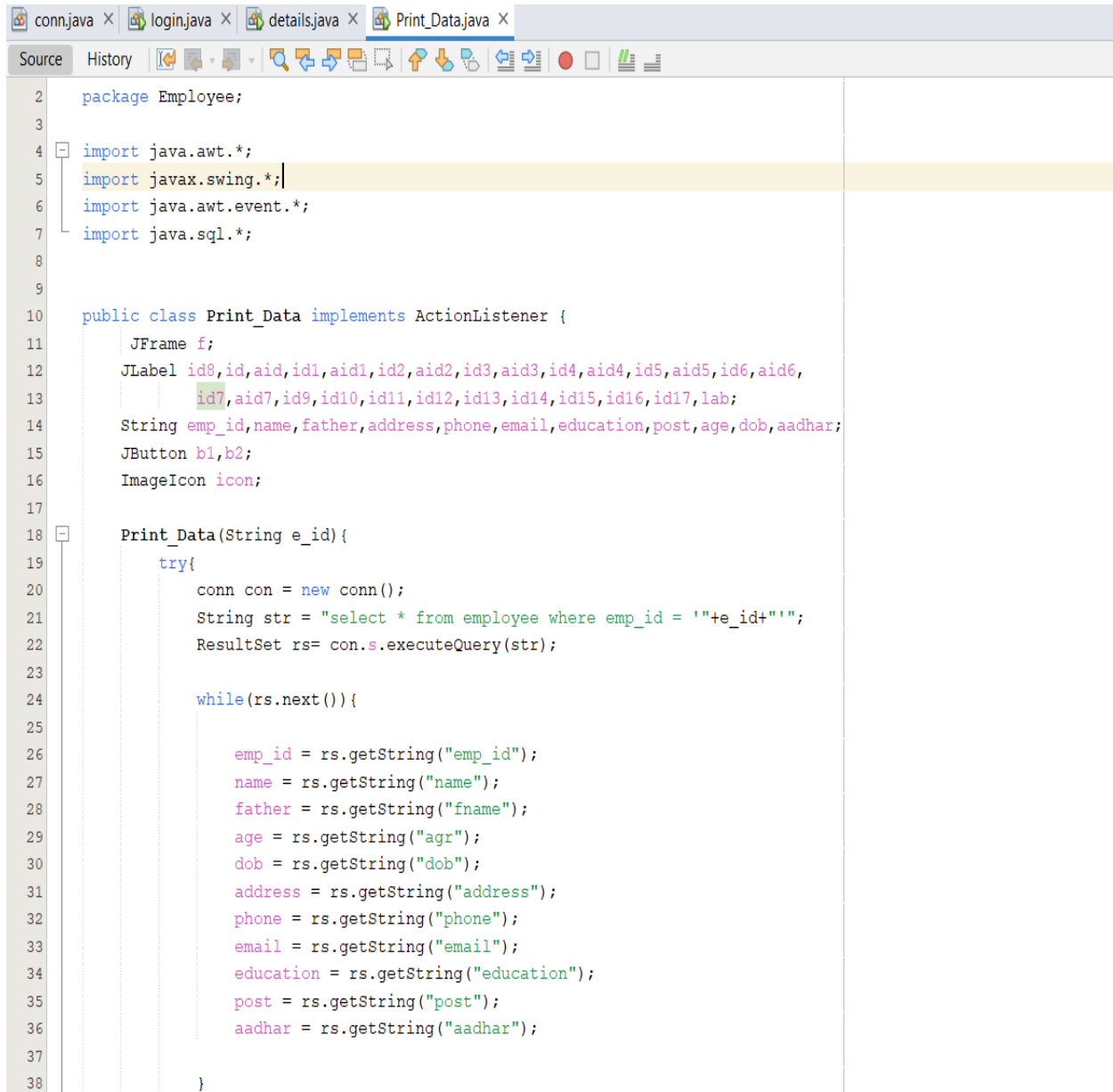
```
conn.java x login.java x details.java x Add_Employee.java x View_Employee.java x
Source History
35
36     t=new JTextField();
37     t.setBounds(240,60,220,30);
38     l1.add(t);
39
40     b=new JButton("Search");
41     b.setBounds(240,150,100,30);
42     b.addActionListener(this);
43     l1.add(b);
44
45     b2=new JButton("Cancel");
46     b2.setBounds(360,150,100,30);
47     b2.addActionListener(this);
48     l1.add(b2);
49
50     f.setSize(500,270);
51     f.setLocation(450,250);
52     f.setVisible(true);
53 }
54
55 public void actionPerformed(ActionEvent ae){
56
57     if(ae.getSource()==b2){
58         f.setVisible(false);
59         details d=new details();
60     }
61     if(ae.getSource()==b){
62         f.setVisible(false);
63         Print_Data p=new Print_Data(t.getText());
64     }
65
66 }
67
68 public static void main(String[]ar){
69     View_Employee v=new View_Employee();
70 }
71 }
```

Output:

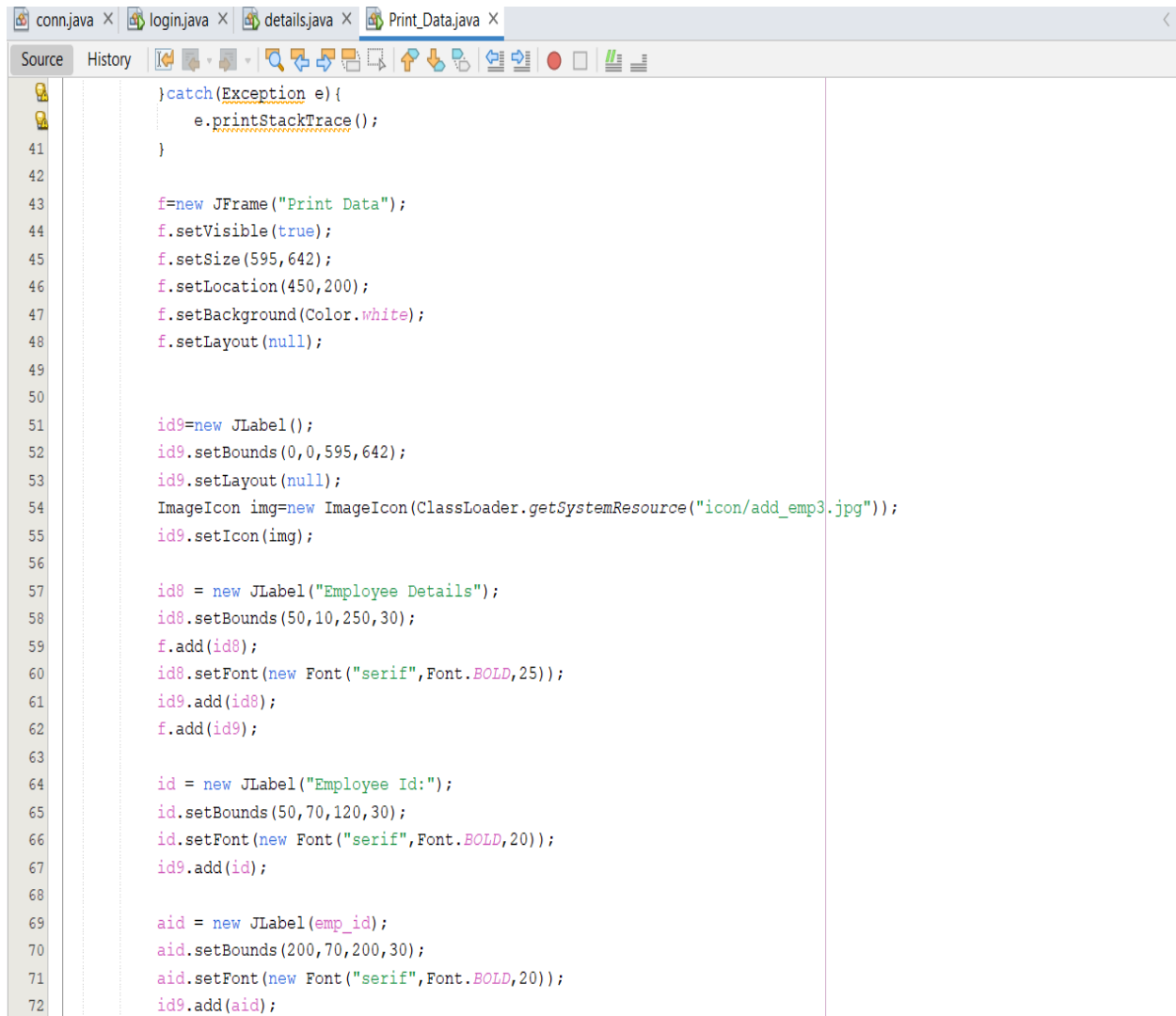


A screenshot of a software window titled "View". The window has a light purple background with a faint pattern of white stars and a large, faint outline of a keyhole. In the center, the text "Employee Id" is displayed in a large, bold, black serif font. To the right of this text is a white rectangular input field containing the number "2049883". Below the input field are two blue rectangular buttons with white text: "Search" on the left and "Cancel" on the right. The window's title bar includes a small icon of a document with a magnifying glass, the text "View", and standard window control buttons (minimize, maximize, close).

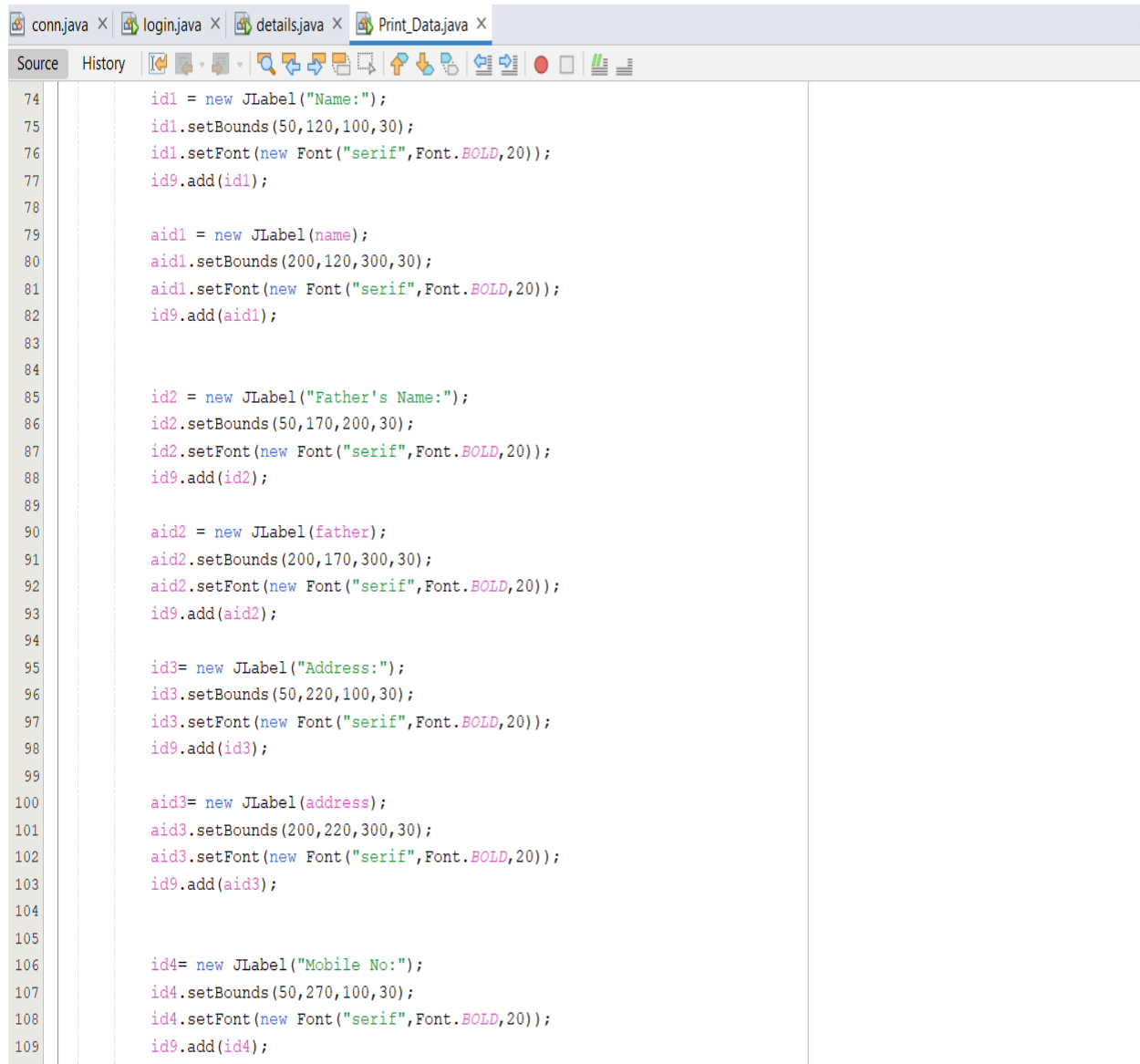
Class:[Print_Data.java]



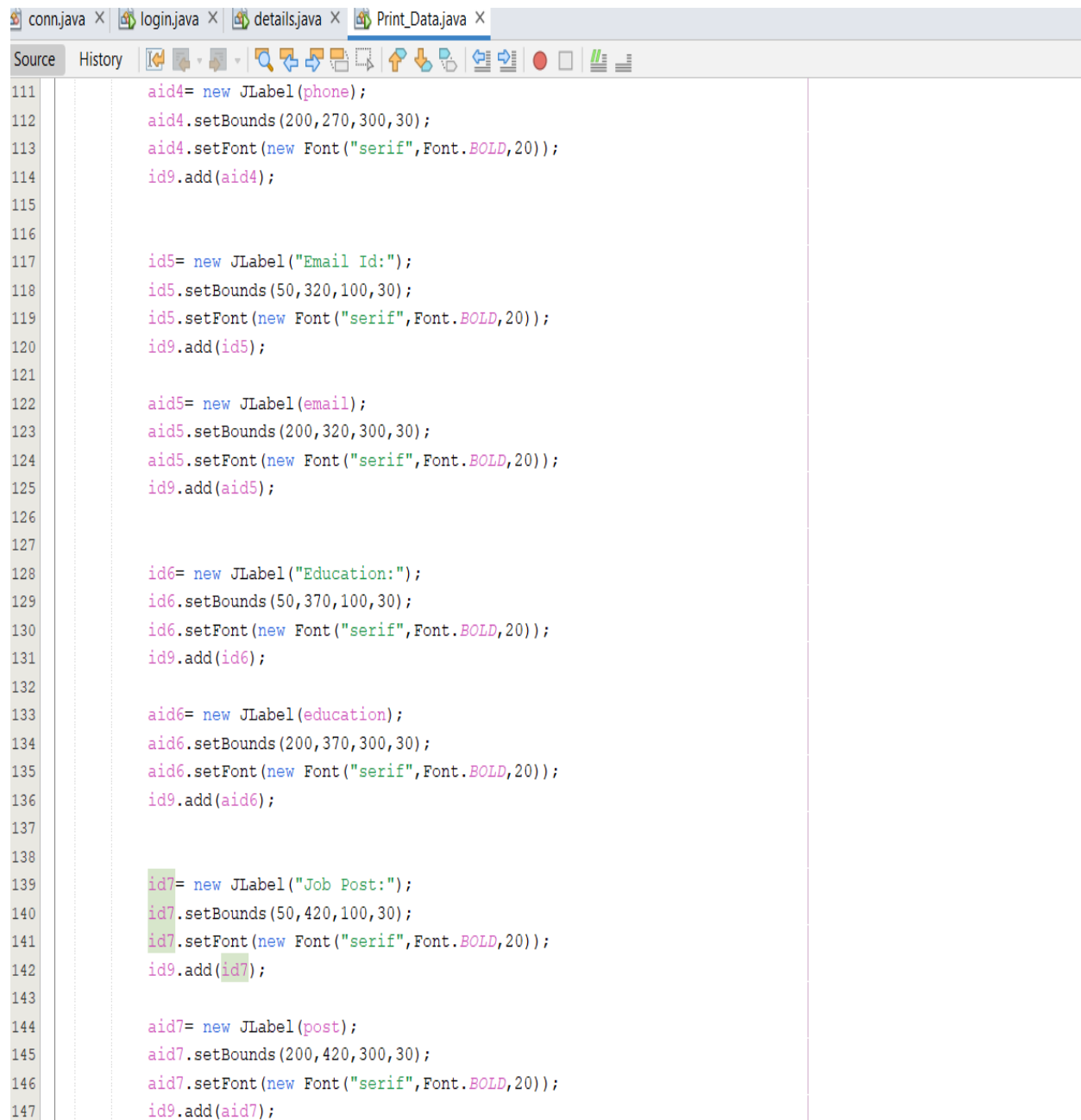
```
2 package Employee;
3
4 import java.awt.*;
5 import javax.swing.*;
6 import java.awt.event.*;
7 import java.sql.*;
8
9
10 public class Print_Data implements ActionListener {
11     JFrame f;
12     JLabel id8, id, aid, id1, aid1, id2, aid2, id3, aid3, id4, aid4, id5, aid5, id6, aid6,
13         id7, aid7, id9, id10, id11, id12, id13, id14, id15, id16, id17, lab;
14     String emp_id, name, father, address, phone, email, education, post, age, dob, aadhar;
15     JButton b1, b2;
16     ImageIcon icon;
17
18     Print_Data(String e_id) {
19         try {
20             Conn con = new Conn();
21             String str = "select * from employee where emp_id = '"+e_id+"'";
22             ResultSet rs = con.s.executeQuery(str);
23
24             while(rs.next()) {
25
26                 emp_id = rs.getString("emp_id");
27                 name = rs.getString("name");
28                 father = rs.getString("fname");
29                 age = rs.getString("agr");
30                 dob = rs.getString("dob");
31                 address = rs.getString("address");
32                 phone = rs.getString("phone");
33                 email = rs.getString("email");
34                 education = rs.getString("education");
35                 post = rs.getString("post");
36                 aadhar = rs.getString("aadhar");
37
38             }
```



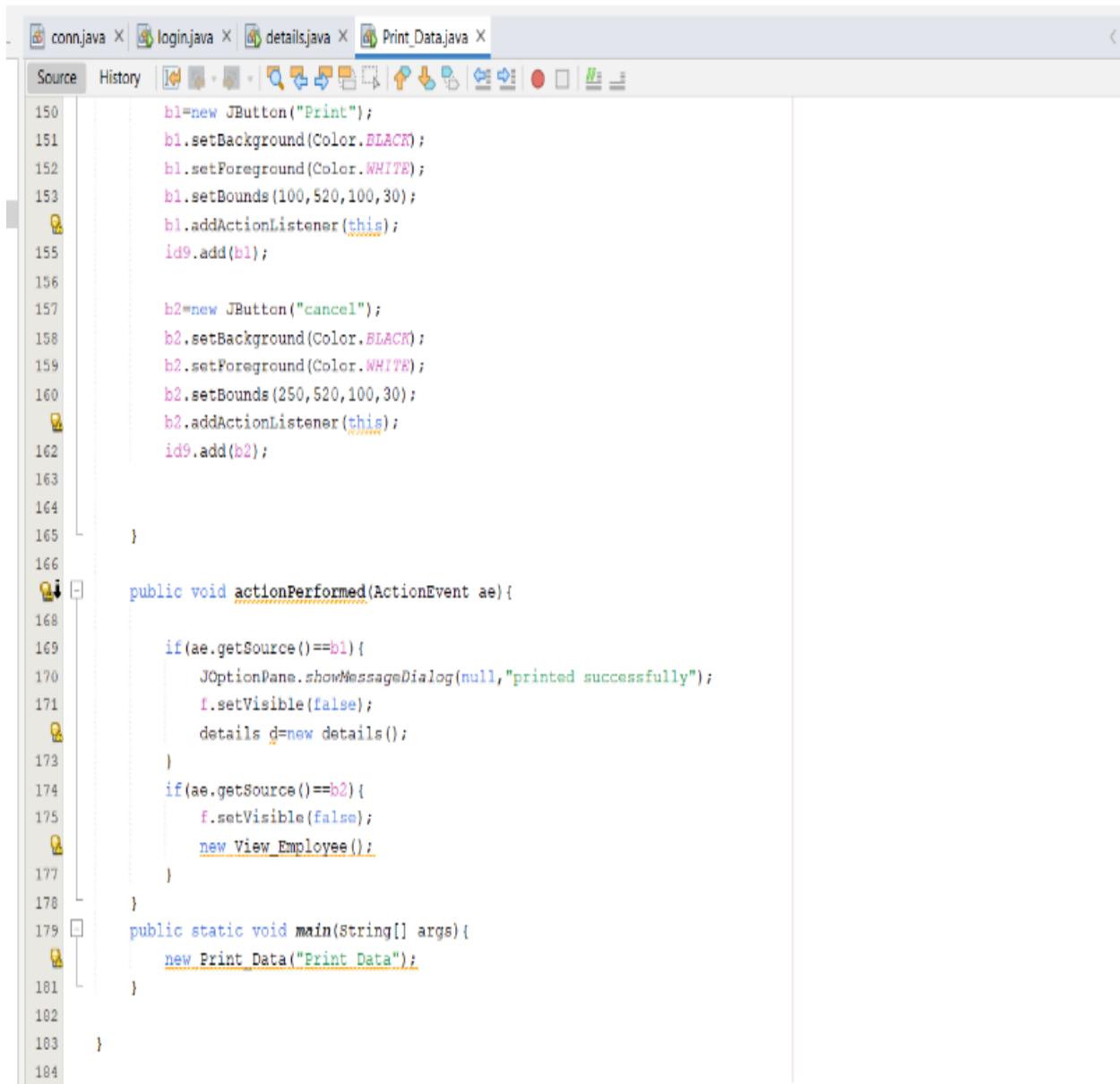
```
conn.java x login.java x details.java x Print_Data.java x
Source History
}catch(Exception e){
    e.printStackTrace();
}
41
42
43     f=new JFrame("Print Data");
44     f.setVisible(true);
45     f.setSize(595,642);
46     f.setLocation(450,200);
47     f.setBackground(Color.white);
48     f.setLayout(null);
49
50
51     id9=new JLabel();
52     id9.setBounds(0,0,595,642);
53     id9.setLayout(null);
54     ImageIcon img=new ImageIcon(ClassLoader.getResource("icon/add_emp3.jpg"));
55     id9.setIcon(img);
56
57     id8 = new JLabel("Employee Details");
58     id8.setBounds(50,10,250,30);
59     f.add(id8);
60     id8.setFont(new Font("serif",Font.BOLD,25));
61     id9.add(id8);
62     f.add(id9);
63
64     id = new JLabel("Employee Id:");
65     id.setBounds(50,70,120,30);
66     id.setFont(new Font("serif",Font.BOLD,20));
67     id9.add(id);
68
69     aid = new JLabel(emp_id);
70     aid.setBounds(200,70,200,30);
71     aid.setFont(new Font("serif",Font.BOLD,20));
72     id9.add(aid);
```

A screenshot of an IDE window with four tabs: conn.java, login.java, details.java, and Print_Data.java. The Print_Data.java tab is active, showing Java code for creating and configuring JLabels. The code is organized into five groups, each with a label ID (id1, id2, id3, id4) and an associated action ID (aid1, aid2, aid3, aid4). Each label is created with a specific text, bounds, and font (serif, bold, size 20). The labels are then added to a container named id9. The code is as follows:

```
74 id1 = new JLabel("Name:");
75 id1.setBounds(50,120,100,30);
76 id1.setFont(new Font("serif",Font.BOLD,20));
77 id9.add(id1);
78
79 aid1 = new JLabel(name);
80 aid1.setBounds(200,120,300,30);
81 aid1.setFont(new Font("serif",Font.BOLD,20));
82 id9.add(aid1);
83
84
85 id2 = new JLabel("Father's Name:");
86 id2.setBounds(50,170,200,30);
87 id2.setFont(new Font("serif",Font.BOLD,20));
88 id9.add(id2);
89
90 aid2 = new JLabel(father);
91 aid2.setBounds(200,170,300,30);
92 aid2.setFont(new Font("serif",Font.BOLD,20));
93 id9.add(aid2);
94
95 id3= new JLabel("Address:");
96 id3.setBounds(50,220,100,30);
97 id3.setFont(new Font("serif",Font.BOLD,20));
98 id9.add(id3);
99
100 aid3= new JLabel(address);
101 aid3.setBounds(200,220,300,30);
102 aid3.setFont(new Font("serif",Font.BOLD,20));
103 id9.add(aid3);
104
105
106 id4= new JLabel("Mobile No:");
107 id4.setBounds(50,270,100,30);
108 id4.setFont(new Font("serif",Font.BOLD,20));
109 id9.add(id4);
```



```
111 aid4= new JLabel(phone);
112 aid4.setBounds(200,270,300,30);
113 aid4.setFont(new Font("serif",Font.BOLD,20));
114 id9.add(aid4);
115
116
117 id5= new JLabel("Email Id:");
118 id5.setBounds(50,320,100,30);
119 id5.setFont(new Font("serif",Font.BOLD,20));
120 id9.add(id5);
121
122 aid5= new JLabel(email);
123 aid5.setBounds(200,320,300,30);
124 aid5.setFont(new Font("serif",Font.BOLD,20));
125 id9.add(aid5);
126
127
128 id6= new JLabel("Education:");
129 id6.setBounds(50,370,100,30);
130 id6.setFont(new Font("serif",Font.BOLD,20));
131 id9.add(id6);
132
133 aid6= new JLabel(education);
134 aid6.setBounds(200,370,300,30);
135 aid6.setFont(new Font("serif",Font.BOLD,20));
136 id9.add(aid6);
137
138
139 id7= new JLabel("Job Post:");
140 id7.setBounds(50,420,100,30);
141 id7.setFont(new Font("serif",Font.BOLD,20));
142 id9.add(id7);
143
144 aid7= new JLabel(post);
145 aid7.setBounds(200,420,300,30);
146 aid7.setFont(new Font("serif",Font.BOLD,20));
147 id9.add(aid7);
```



```
150         b1=new JButton("Print");
151         b1.setBackground(Color.BLACK);
152         b1.setForeground(Color.WHITE);
153         b1.setBounds(100,520,100,30);
154         b1.addActionListener(this);
155         id9.add(b1);
156
157         b2=new JButton("cancel");
158         b2.setBackground(Color.BLACK);
159         b2.setForeground(Color.WHITE);
160         b2.setBounds(250,520,100,30);
161         b2.addActionListener(this);
162         id9.add(b2);
163
164     }
165
166     public void actionPerformed(ActionEvent ae){
167
168         if(ae.getSource()==b1){
169             JOptionPane.showMessageDialog(null,"printed successfully");
170             f.setVisible(false);
171             details d=new details();
172         }
173         if(ae.getSource()==b2){
174             f.setVisible(false);
175             new View_Employee();
176         }
177     }
178
179     public static void main(String[] args){
180         new Print_Data("Print Data");
181     }
182
183 }
184
```

Output:

 Print Data

— □ ×



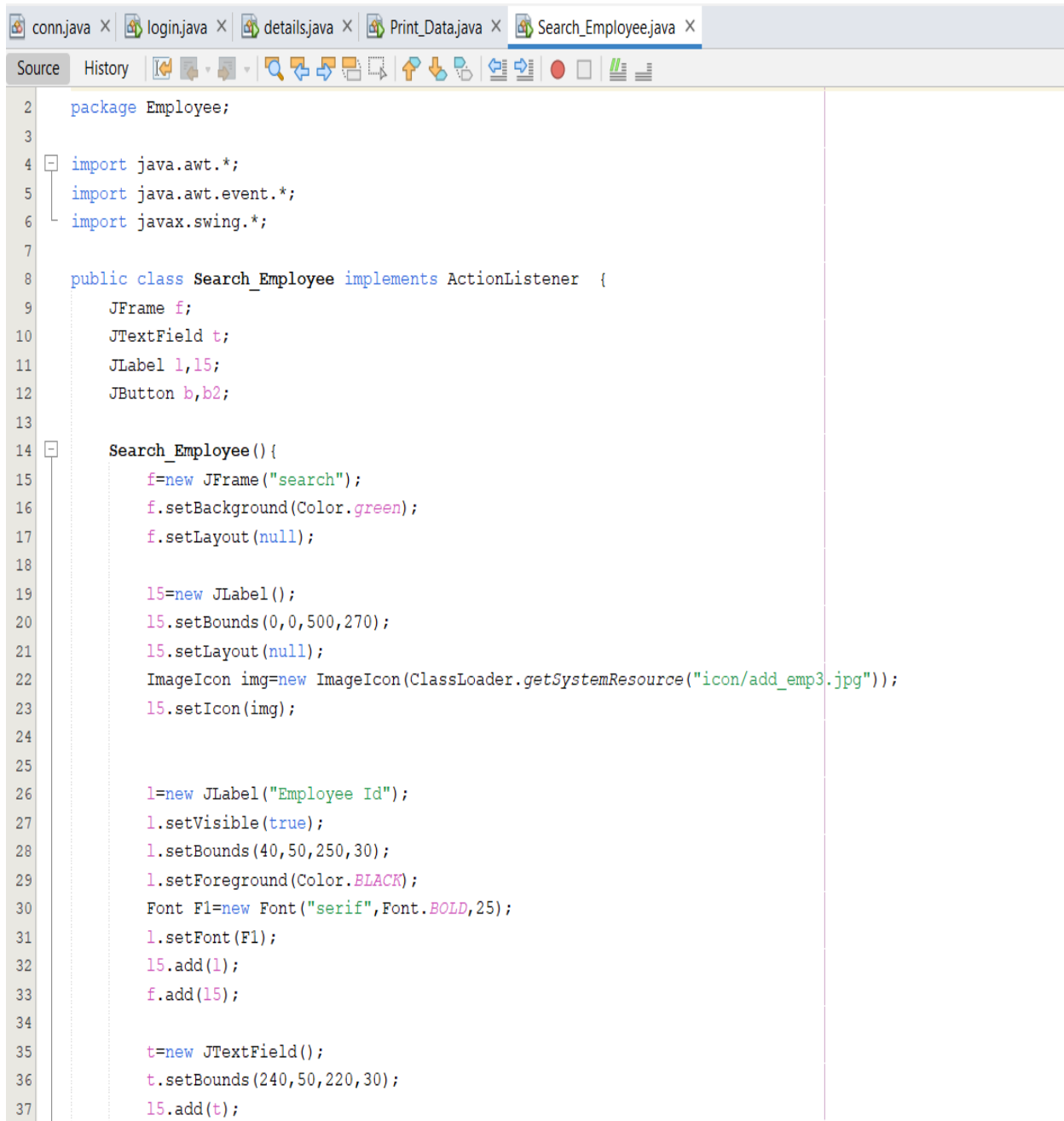
Employee Details

Employee Id:	2049883
Name:	Dipika Ray
Father's Name:	Alok Ray
Address:	Kolkata
Mobile No:	82407323
Email Id:	dipikaray52@gmail.com
Education:	Msc
Job Post:	Trainee

Print

cancel

Class:[Search_Employee.java]



```
2 package Employee;
3
4 import java.awt.*;
5 import java.awt.event.*;
6 import javax.swing.*;
7
8 public class Search_Employee implements ActionListener {
9     JFrame f;
10    JTextField t;
11    JLabel l,l5;
12    JButton b,b2;
13
14    Search_Employee(){
15        f=new JFrame("search");
16        f.setBackground(Color.green);
17        f.setLayout(null);
18
19        l5=new JLabel();
20        l5.setBounds(0,0,500,270);
21        l5.setLayout(null);
22        ImageIcon img=new ImageIcon(ClassLoader.getResource("icon/add_emp3.jpg"));
23        l5.setIcon(img);
24
25
26        l=new JLabel("Employee Id");
27        l.setVisible(true);
28        l.setBounds(40,50,250,30);
29        l.setForeground(Color.BLACK);
30        Font f1=new Font("serif",Font.BOLD,25);
31        l.setFont(f1);
32        l5.add(l);
33        f.add(l5);
34
35        t=new JTextField();
36        t.setBounds(240,50,220,30);
37        l5.add(t);
```

```

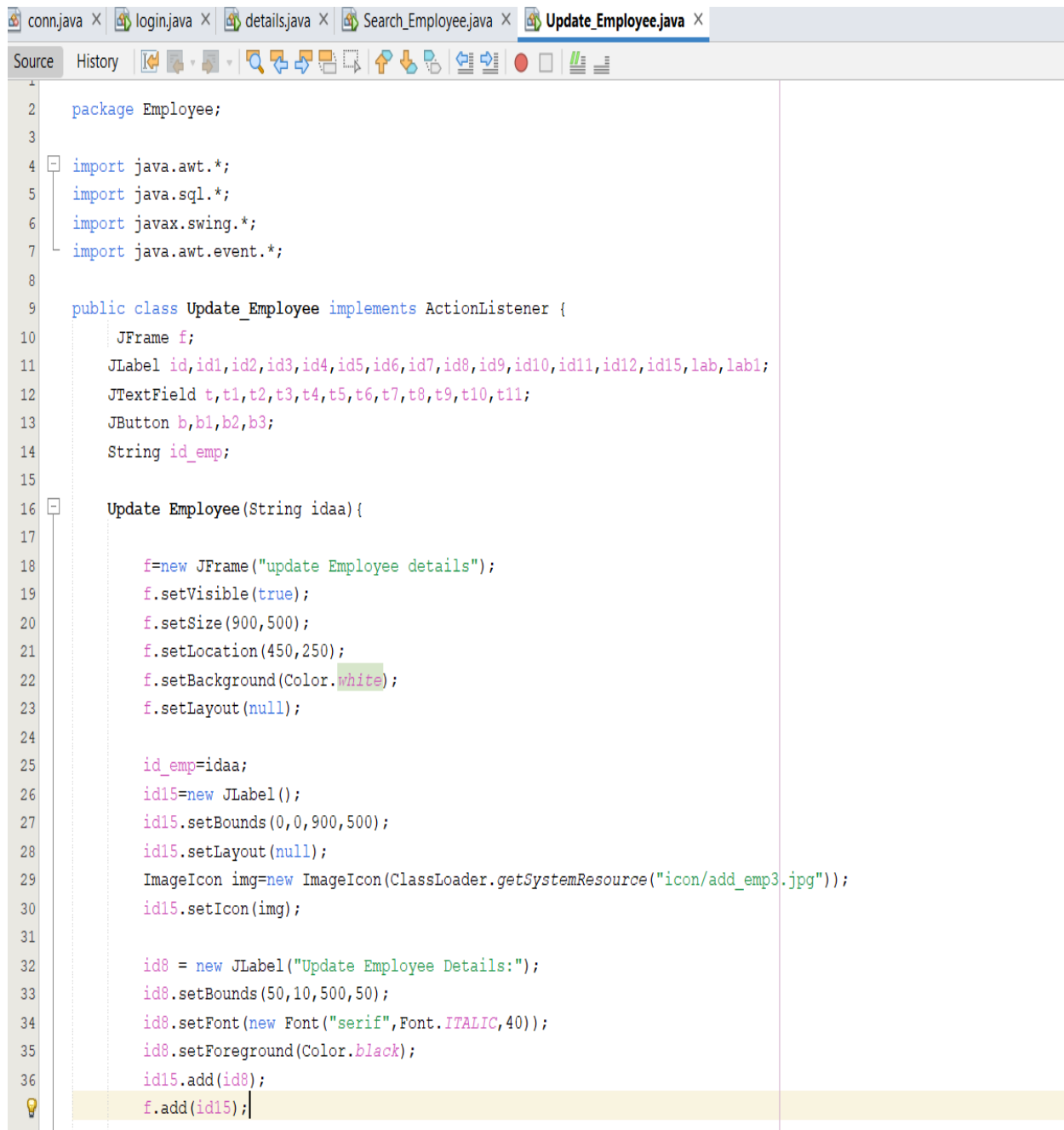
conn.java x login.java x details.java x Print_Data.java x Search_Employee.java x
Source History
39      b=new JButton("Search");
40      b.setBounds(240,150,100,30);
41      b.addActionListener(this);
42      f.add(b);
43
44
45      b2=new JButton("Cancel");
46      b2.setBounds(360,150,100,30);
47      b2.addActionListener(this);
48      f.add(b2);
49
50      f.setSize(500,270);
51      f.setLocation(450,250);
52      f.setVisible(true);
53  }
54
55  public void actionPerformed(ActionEvent ae){
56      if(ae.getSource()==b2){
57          f.setVisible(false);
58          details d=new details();
59      }
60      if(ae.getSource()==b){
61          new Update_Employee(t.getText());
62          f.setVisible(false);
63      }
64  }
65  }
66
67  public static void main(String[] arg){
68      Search_Employee obj =new Search_Employee();
69  }
70
71  }

```

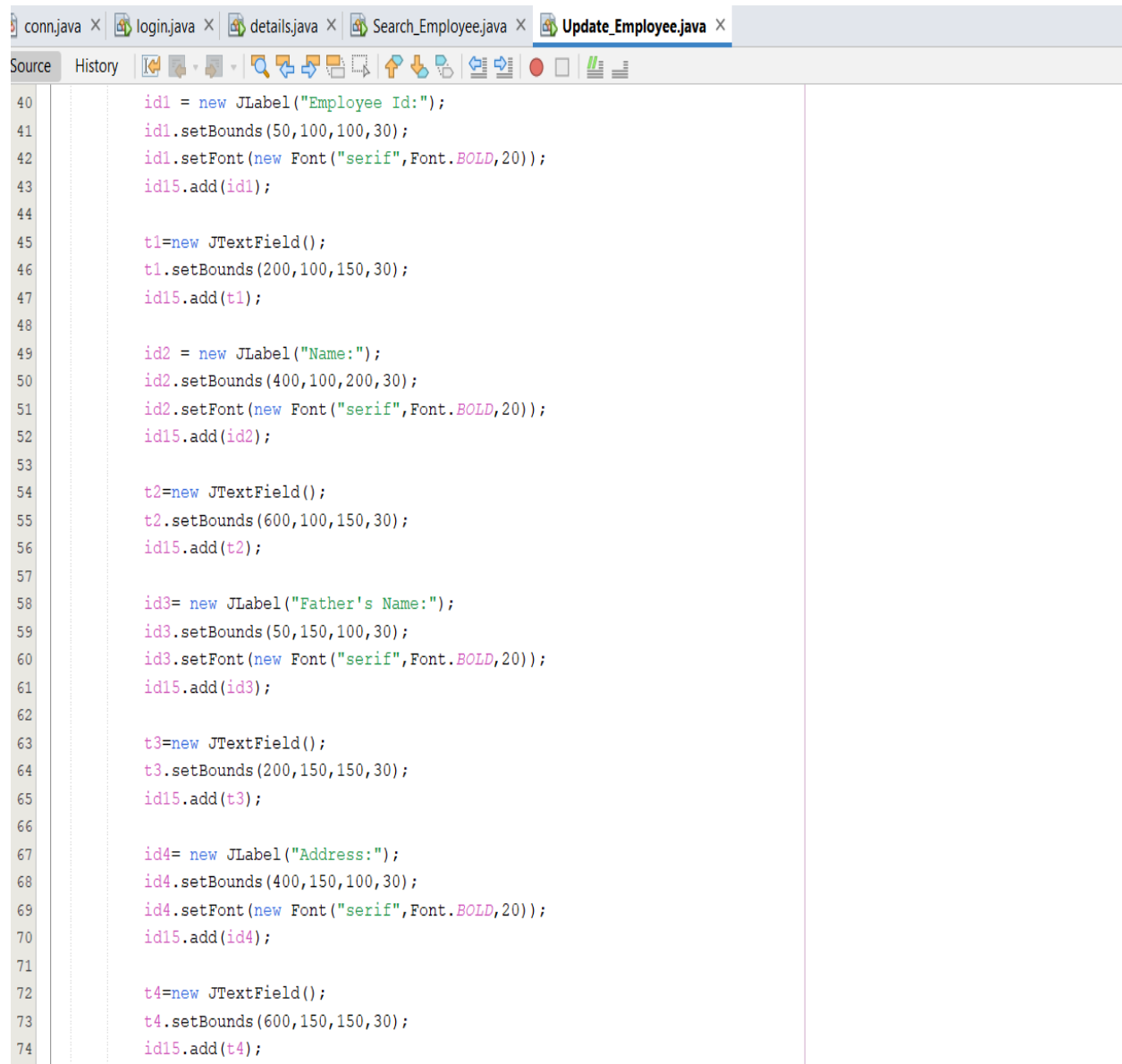
Output:



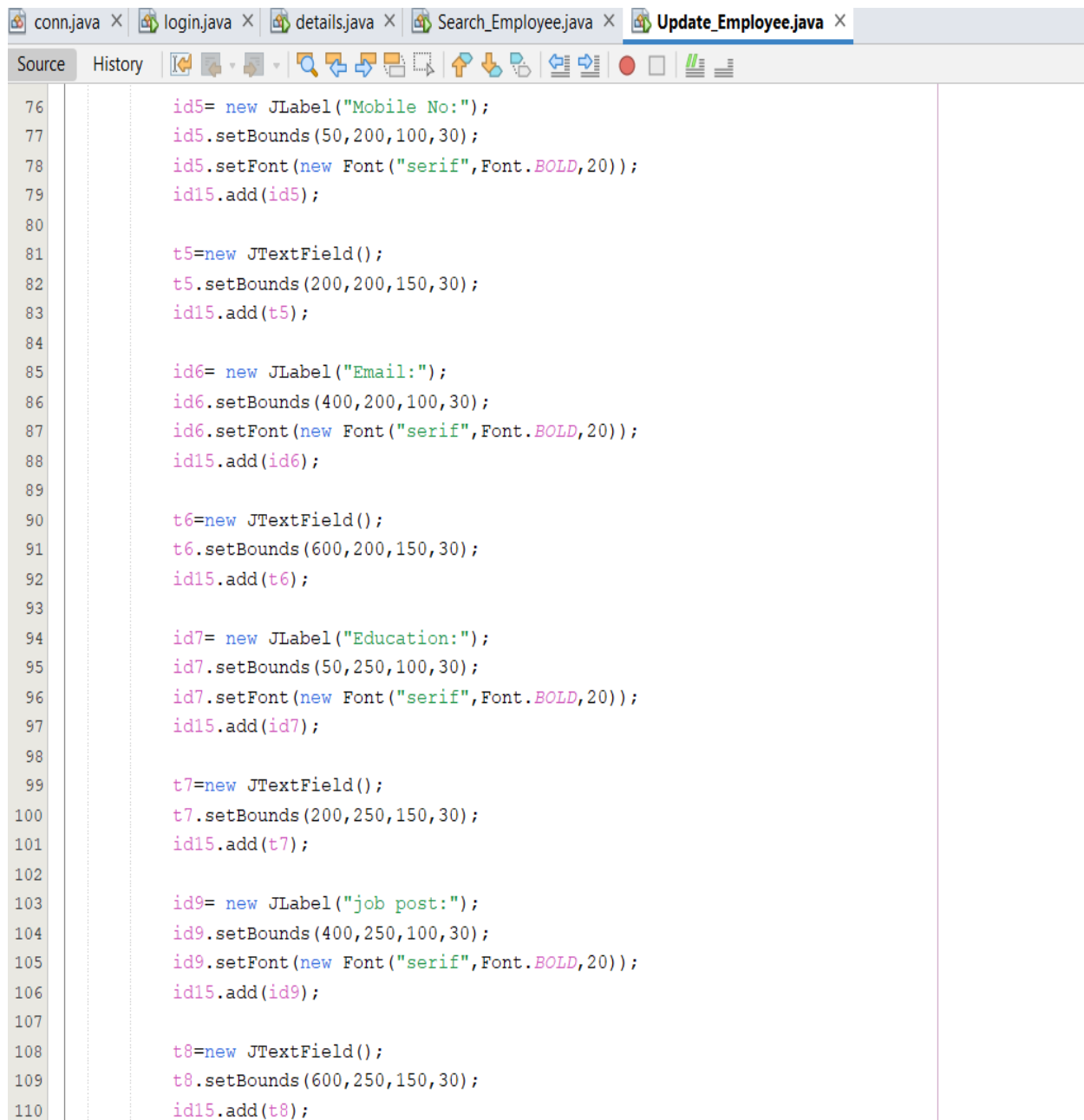
Class:[Update_Employee.java]



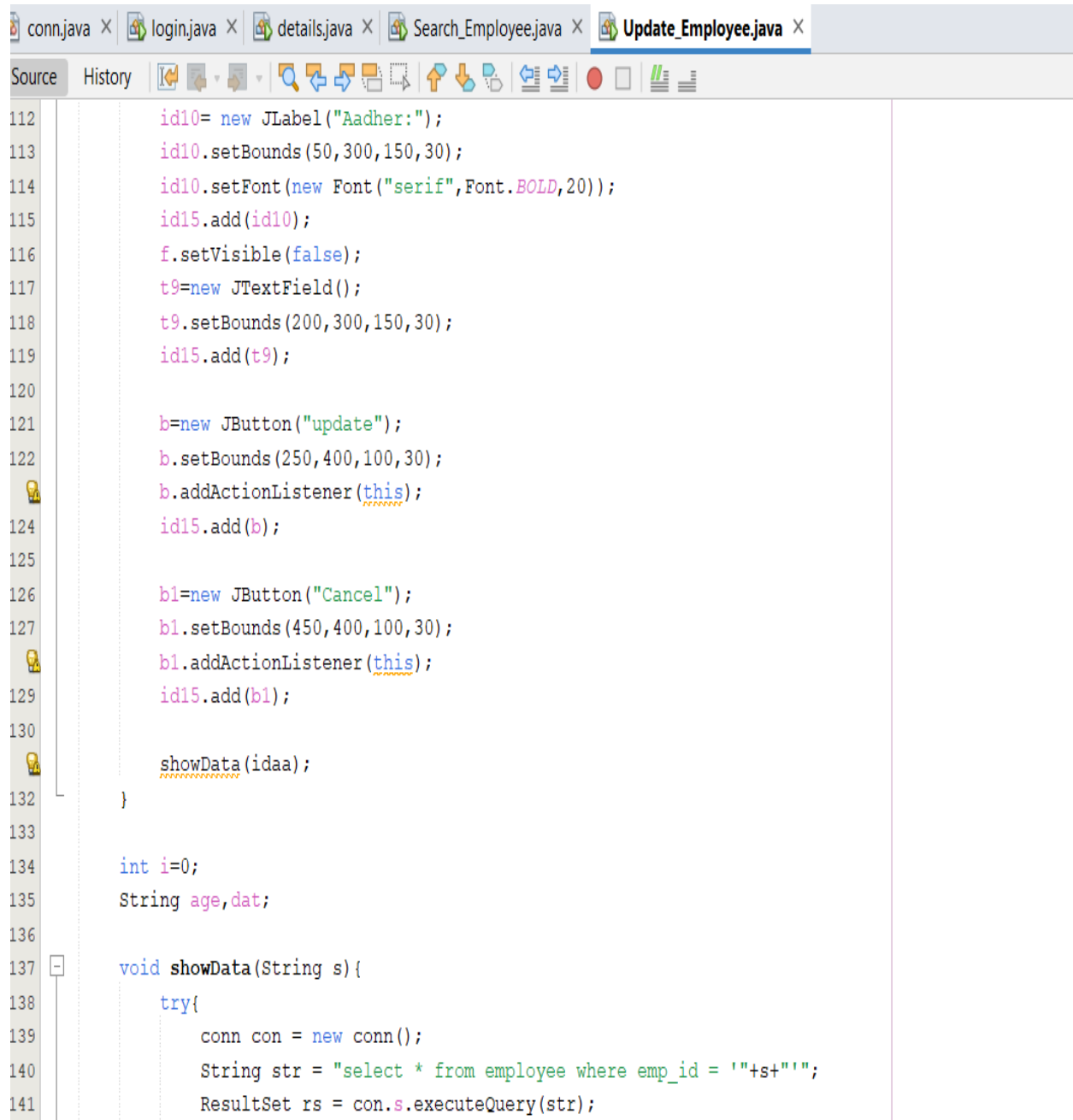
```
1
2 package Employee;
3
4 import java.awt.*;
5 import java.sql.*;
6 import javax.swing.*;
7 import java.awt.event.*;
8
9 public class Update_Employee implements ActionListener {
10     JFrame f;
11     JLabel id, id1, id2, id3, id4, id5, id6, id7, id8, id9, id10, id11, id12, id15, lab, lab1;
12     JTextField t, t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11;
13     JButton b, b1, b2, b3;
14     String id_emp;
15
16     Update_Employee(String idaa) {
17
18         f=new JFrame("update Employee details");
19         f.setVisible(true);
20         f.setSize(900,500);
21         f.setLocation(450,250);
22         f.setBackground(Color.white);
23         f.setLayout(null);
24
25         id_emp=idaa;
26         id15=new JLabel();
27         id15.setBounds(0,0,900,500);
28         id15.setLayout(null);
29         ImageIcon img=new ImageIcon(ClassLoader.getResource("icon/add_emp3.jpg"));
30         id15.setIcon(img);
31
32         id8 = new JLabel("Update Employee Details:");
33         id8.setBounds(50,10,500,50);
34         id8.setFont(new Font("serif",Font.ITALIC,40));
35         id8.setForeground(Color.black);
36         id15.add(id8);
37         f.add(id15);
38     }
39 }
```



```
40     id1 = new JLabel("Employee Id:");
41     id1.setBounds(50,100,100,30);
42     id1.setFont(new Font("serif",Font.BOLD,20));
43     id15.add(id1);
44
45     t1=new JTextField();
46     t1.setBounds(200,100,150,30);
47     id15.add(t1);
48
49     id2 = new JLabel("Name:");
50     id2.setBounds(400,100,200,30);
51     id2.setFont(new Font("serif",Font.BOLD,20));
52     id15.add(id2);
53
54     t2=new JTextField();
55     t2.setBounds(600,100,150,30);
56     id15.add(t2);
57
58     id3= new JLabel("Father's Name:");
59     id3.setBounds(50,150,100,30);
60     id3.setFont(new Font("serif",Font.BOLD,20));
61     id15.add(id3);
62
63     t3=new JTextField();
64     t3.setBounds(200,150,150,30);
65     id15.add(t3);
66
67     id4= new JLabel("Address:");
68     id4.setBounds(400,150,100,30);
69     id4.setFont(new Font("serif",Font.BOLD,20));
70     id15.add(id4);
71
72     t4=new JTextField();
73     t4.setBounds(600,150,150,30);
74     id15.add(t4);
```



```
76     id5= new JLabel("Mobile No:");
77     id5.setBounds(50,200,100,30);
78     id5.setFont(new Font("serif",Font.BOLD,20));
79     id15.add(id5);
80
81     t5=new JTextField();
82     t5.setBounds(200,200,150,30);
83     id15.add(t5);
84
85     id6= new JLabel("Email:");
86     id6.setBounds(400,200,100,30);
87     id6.setFont(new Font("serif",Font.BOLD,20));
88     id15.add(id6);
89
90     t6=new JTextField();
91     t6.setBounds(600,200,150,30);
92     id15.add(t6);
93
94     id7= new JLabel("Education:");
95     id7.setBounds(50,250,100,30);
96     id7.setFont(new Font("serif",Font.BOLD,20));
97     id15.add(id7);
98
99     t7=new JTextField();
100    t7.setBounds(200,250,150,30);
101    id15.add(t7);
102
103    id9= new JLabel("job post:");
104    id9.setBounds(400,250,100,30);
105    id9.setFont(new Font("serif",Font.BOLD,20));
106    id15.add(id9);
107
108    t8=new JTextField();
109    t8.setBounds(600,250,150,30);
110    id15.add(t8);
```



```
conn.java × login.java × details.java × Search_Employee.java × Update_Employee.java ×
Source History
112     id10= new JLabel("Aadher:");
113     id10.setBounds(50,300,150,30);
114     id10.setFont(new Font("serif",Font.BOLD,20));
115     id15.add(id10);
116     f.setVisible(false);
117     t9=new JTextField();
118     t9.setBounds(200,300,150,30);
119     id15.add(t9);
120
121     b=new JButton("update");
122     b.setBounds(250,400,100,30);
123     b.addActionListener(this);
124     id15.add(b);
125
126     b1=new JButton("Cancel");
127     b1.setBounds(450,400,100,30);
128     b1.addActionListener(this);
129     id15.add(b1);
130
131     showData(idaa);
132 }
133
134 int i=0;
135 String age,dat;
136
137 void showData(String s){
138     try{
139         conn con = new conn();
140         String str = "select * from employee where emp_id = '"+s+"'";
141         ResultSet rs = con.s.executeQuery(str);
```

```
conn.java x login.java x details.java x Search_Employee.java x Update_Employee.java x
Source History
143         if(rs.next()) {
144             f.setVisible(true);
145             i=1;
146
147             t1.setText(rs.getString(1));
148             t2.setText(rs.getString(2));
149             t3.setText(rs.getString(3));
150             t4.setText(rs.getString(6));
151             t5.setText(rs.getString(7));
152             t6.setText(rs.getString(8));
153             t7.setText(rs.getString(9));
154             t8.setText(rs.getString(10));
155             t9.setText(rs.getString(11));
156
157             age=rs.getString(4);
158             dat=rs.getString(5);
159         }
160         if(i==0)
161             JOptionPane.showMessageDialog(null,"Id not found");
162         new Search_Employee();
163     }catch(Exception ex){}
164     f.setVisible(true);
165     f.setSize(900,500);
166     f.setLocation(400,100);
167 }
```

```
conn.java X login.java X details.java X Search_Employee.java X Update_Employee.java X
Source History
166 public void actionPerformed(ActionEvent ae){
170     if(ae.getSource()==b && i==1){
171         try{
172             conn con = new conn();
173             String str = "update employee set emp_id='"+t1.getText()+"',name='"+t2.getText()+"', "
174                 + "fname='"+t3.getText()+"',agr='"+age+"',dob='"+dat+"',address='"+t4.getText()+"', "
175                 + "phone='"+t5.getText()+"',email='"+t6.getText()+"',education='"+t7.getText()+"' "
176                 + "post='"+t8.getText()+"',aadhar='"+t9.getText()+"' where emp_id='"+id_emp+"'";
177             con.s.executeUpdate(str);
178             JOptionPane.showMessageDialog(null,"successfully updated");
179             f.setVisible(false);
180             new details();
181         }catch(Exception e){
182             System.out.println("The error is:"+e);
183         }
184     }
185     if(ae.getSource()==b1){
186         f.setVisible(false);
187         details d=new details();
188     }
189 }
190
191 public static void main(String[] arg){
192     new Update_Employee("Update Employee");
193 }
194
195 }
196
```


Output:

update Employee details

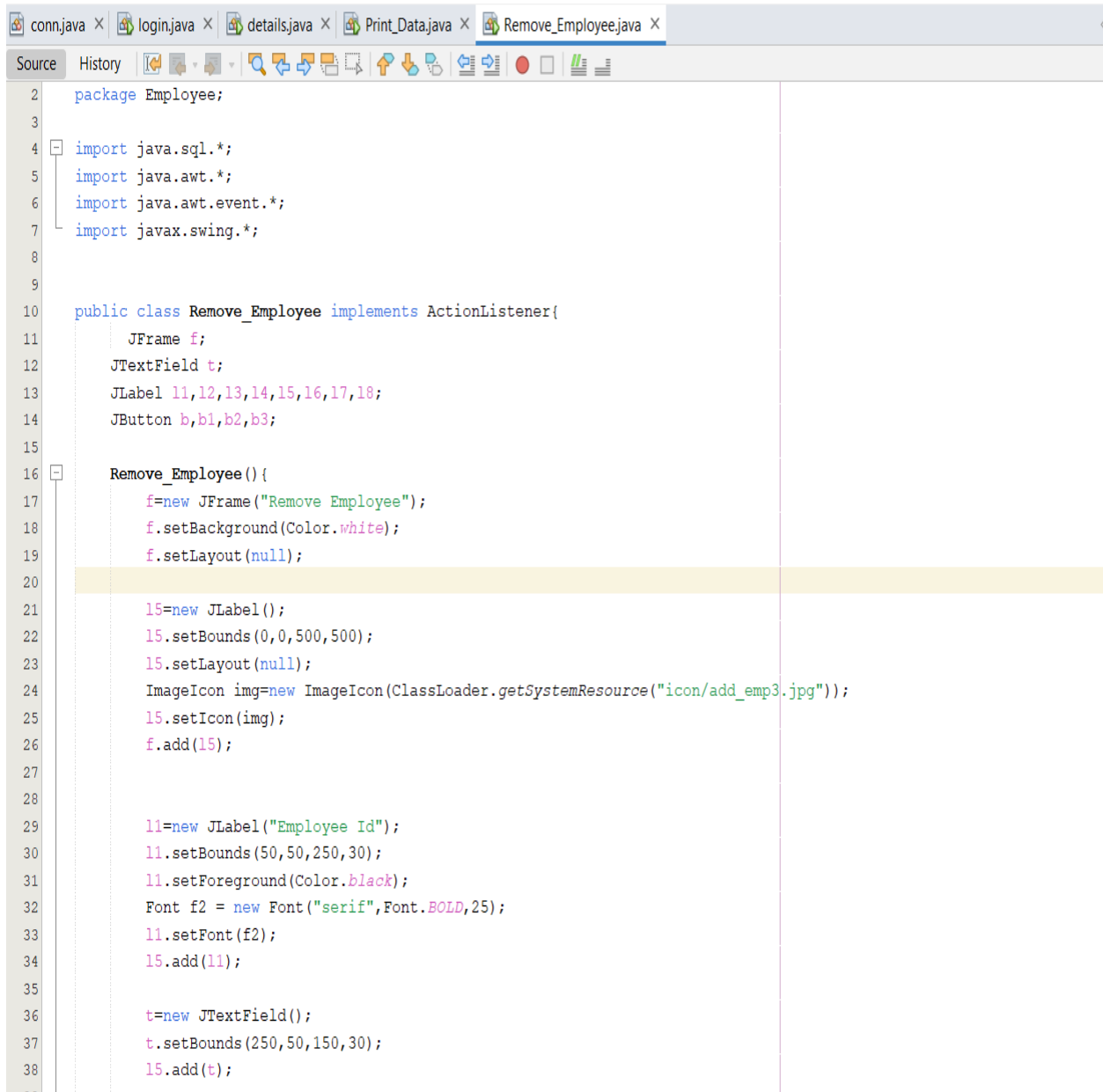
Update Employee Details:

Employee...	4	Name:	Santosh Pal
Father's ...	Hemanta Pal	Address:	Kolkata
Mobile No:	9163292942	Email:	santosh21.sp4@gmail.com
Education:	M.sc	job post:	Trainee
Aadher:	777755552222		

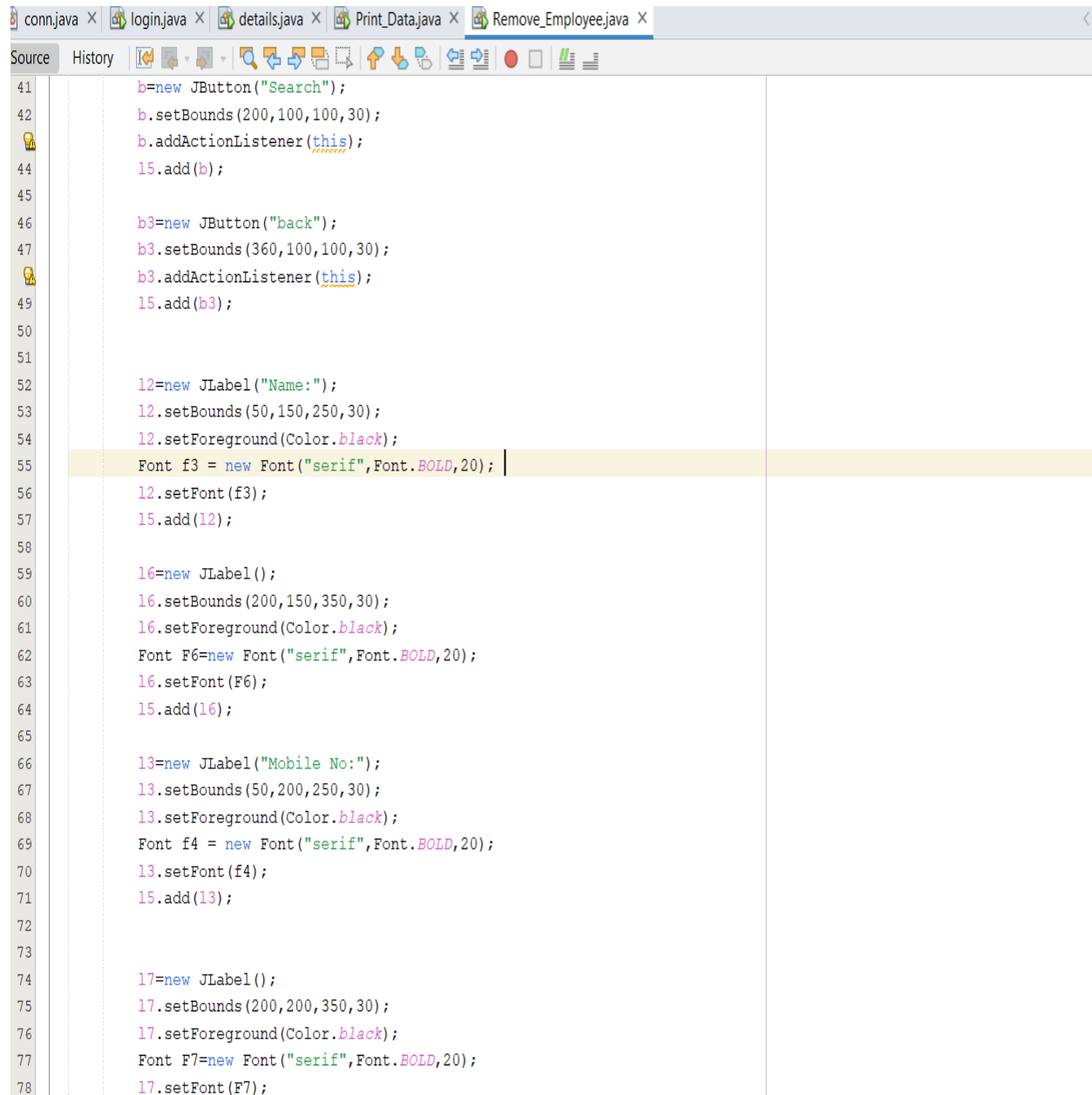
Message

 successfully updated

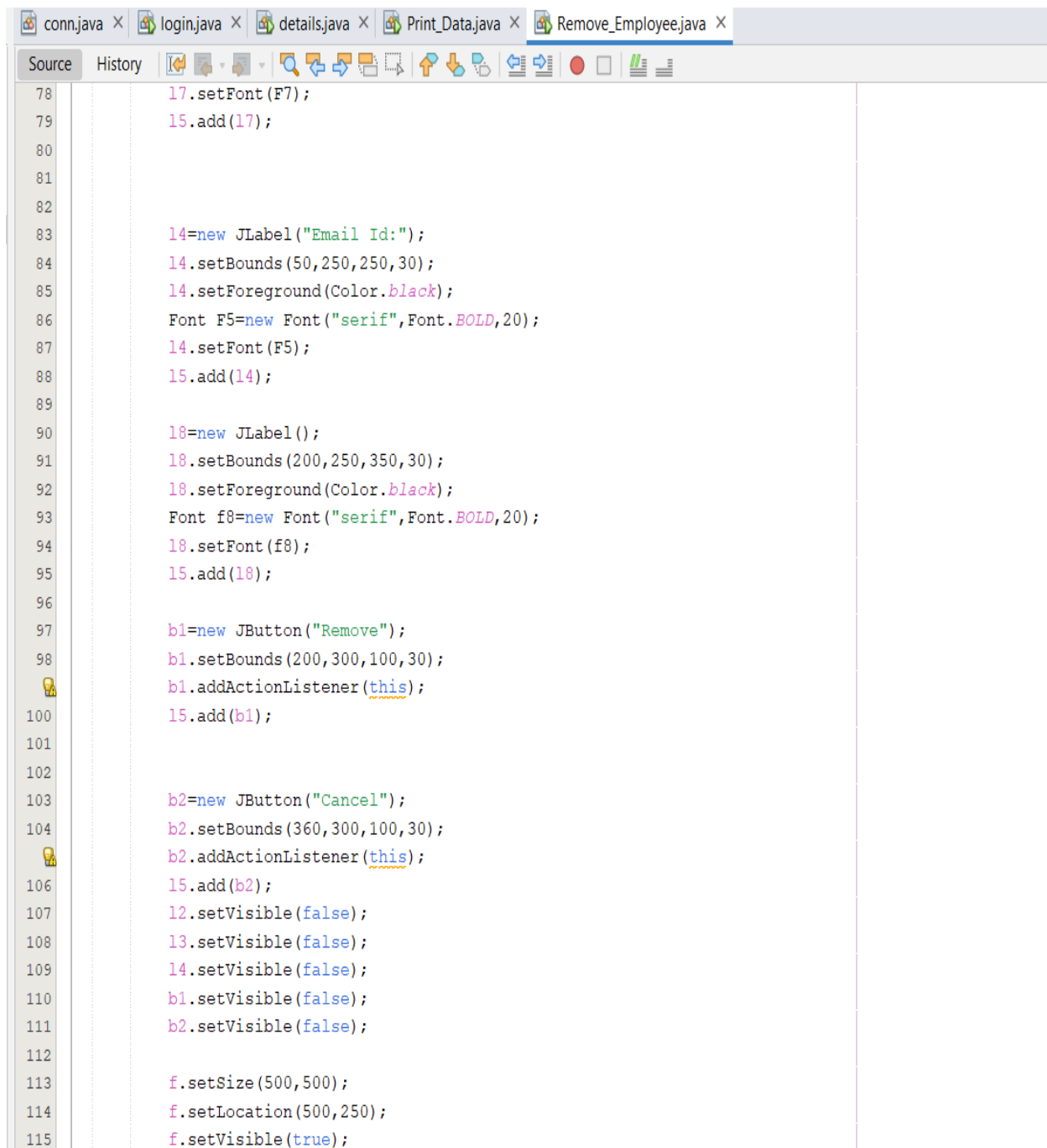
Class:[Remove_Employee.java]



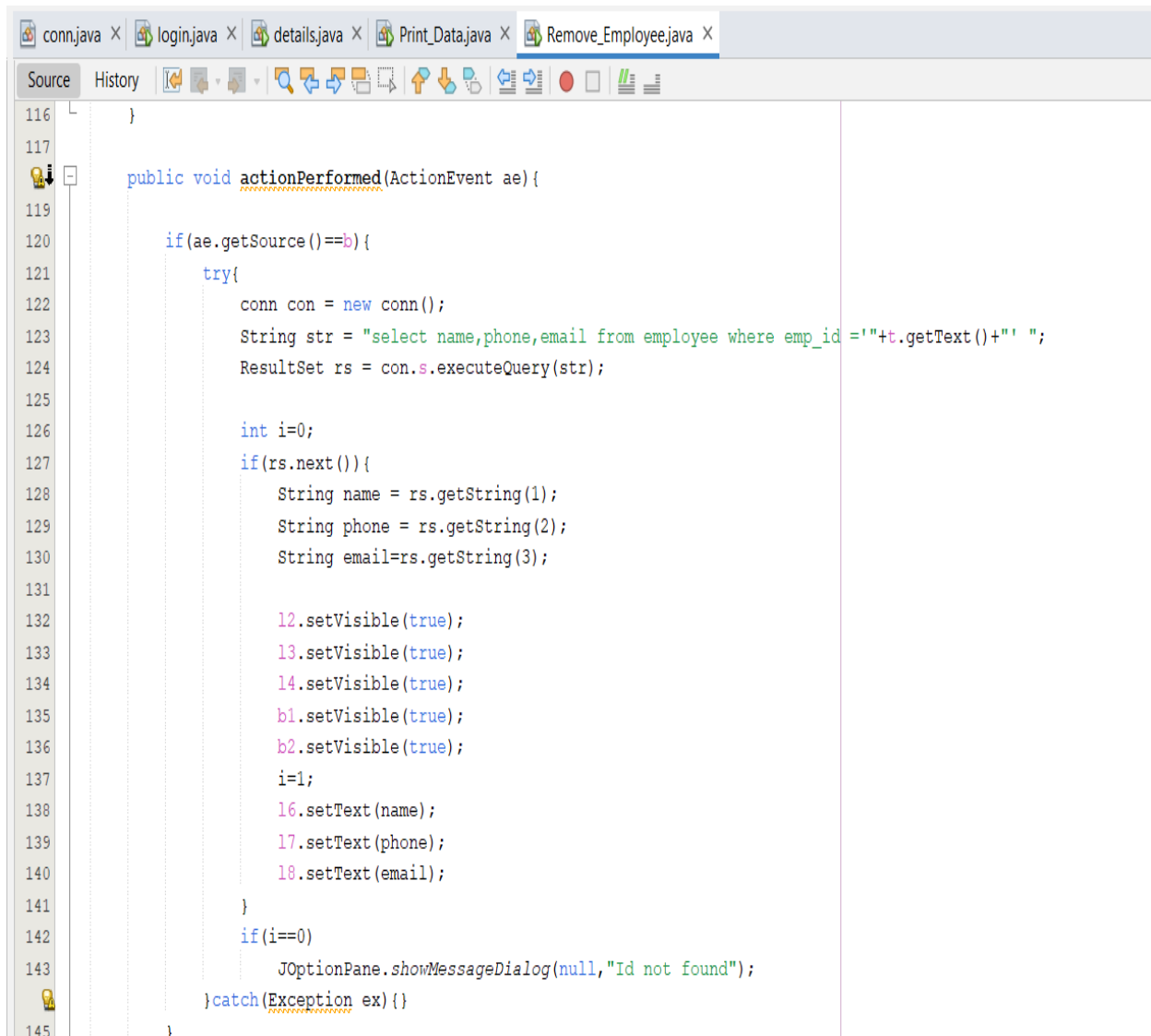
```
2  package Employee;
3
4  import java.sql.*;
5  import java.awt.*;
6  import java.awt.event.*;
7  import javax.swing.*;
8
9
10 public class Remove_Employee implements ActionListener{
11     JFrame f;
12     JTextField t;
13     JLabel l1,l2,l3,l4,l5,l6,l7,l8;
14     JButton b,b1,b2,b3;
15
16     Remove_Employee() {
17         f=new JFrame("Remove Employee");
18         f.setBackground(Color.white);
19         f.setLayout(null);
20
21         l5=new JLabel();
22         l5.setBounds(0,0,500,500);
23         l5.setLayout(null);
24         ImageIcon img=new ImageIcon(ClassLoader.getResource("icon/add_emp3.jpg"));
25         l5.setIcon(img);
26         f.add(l5);
27
28
29         l1=new JLabel("Employee Id");
30         l1.setBounds(50,50,250,30);
31         l1.setForeground(Color.black);
32         Font f2 = new Font("serif",Font.BOLD,25);
33         l1.setFont(f2);
34         l5.add(l1);
35
36         t=new JTextField();
37         t.setBounds(250,50,150,30);
38         l5.add(t);
39     }
40 }
```

```
41      b=new JButton("Search");
42      b.setBounds(200,100,100,30);
43      b.addActionListener(this);
44      l5.add(b);
45
46      b3=new JButton("back");
47      b3.setBounds(360,100,100,30);
48      b3.addActionListener(this);
49      l5.add(b3);
50
51
52      l2=new JLabel("Name:");
53      l2.setBounds(50,150,250,30);
54      l2.setForeground(Color.black);
55      Font f3 = new Font("serif",Font.BOLD,20);
56      l2.setFont(f3);
57      l5.add(l2);
58
59      l6=new JLabel();
60      l6.setBounds(200,150,350,30);
61      l6.setForeground(Color.black);
62      Font F6=new Font("serif",Font.BOLD,20);
63      l6.setFont(F6);
64      l5.add(l6);
65
66      l3=new JLabel("Mobile No:");
67      l3.setBounds(50,200,250,30);
68      l3.setForeground(Color.black);
69      Font f4 = new Font("serif",Font.BOLD,20);
70      l3.setFont(f4);
71      l5.add(l3);
72
73
74      l7=new JLabel();
75      l7.setBounds(200,200,350,30);
76      l7.setForeground(Color.black);
77      Font F7=new Font("serif",Font.BOLD,20);
78      l7.setFont(F7);
```



```
conn.java x login.java x details.java x Print_Data.java x Remove_Employee.java x
Source History
78      17.setFont(F7);
79      15.add(17);
80
81
82
83      14=new JLabel("Email Id:");
84      14.setBounds(50,250,250,30);
85      14.setForeground(Color.black);
86      Font F5=new Font("serif",Font.BOLD,20);
87      14.setFont(F5);
88      15.add(14);
89
90      18=new JLabel();
91      18.setBounds(200,250,350,30);
92      18.setForeground(Color.black);
93      Font f8=new Font("serif",Font.BOLD,20);
94      18.setFont(f8);
95      15.add(18);
96
97      b1=new JButton("Remove");
98      b1.setBounds(200,300,100,30);
99      b1.addActionListener(this);
100     15.add(b1);
101
102
103     b2=new JButton("Cancel");
104     b2.setBounds(360,300,100,30);
105     b2.addActionListener(this);
106     15.add(b2);
107     12.setVisible(false);
108     13.setVisible(false);
109     14.setVisible(false);
110     b1.setVisible(false);
111     b2.setVisible(false);
112
113     f.setSize(500,500);
114     f.setLocation(500,250);
115     f.setVisible(true);
```



```
116     }
117
118     public void actionPerformed(ActionEvent ae){
119
120         if(ae.getSource()==b){
121             try{
122                 conn con = new conn();
123                 String str = "select name,phone,email from employee where emp_id='"+t.getText()+"' ";
124                 ResultSet rs = con.s.executeQuery(str);
125
126                 int i=0;
127                 if(rs.next()){
128                     String name = rs.getString(1);
129                     String phone = rs.getString(2);
130                     String email=rs.getString(3);
131
132                     12.setVisible(true);
133                     13.setVisible(true);
134                     14.setVisible(true);
135                     b1.setVisible(true);
136                     b2.setVisible(true);
137                     i=1;
138                     16.setText(name);
139                     17.setText(phone);
140                     18.setText(email);
141                 }
142                 if(i==0)
143                     JOptionPane.showMessageDialog(null,"Id not found");
144             }catch(Exception ex){}
145         }
```

```
conn.java X login.java X details.java X Print_Data.java X Remove_Employee.java X
Source History
146 if(ae.getSource()==b1){
147     try{
148         conn con = new conn();
149         String str = "delete from employee where emp_id = '"+t.getText()+"'";
150         con.s.executeUpdate(str);
151         JOptionPane.showMessageDialog(null,"deleted successfully");
152         j2.setVisible(false);
153         j3.setVisible(false);
154         j4.setVisible(false);
155         j6.setVisible(false);
156         j7.setVisible(false);
157         j8.setVisible(false);
158         b1.setVisible(false);
159         b2.setVisible(false);
160
161     }catch(Exception ex){
162         JOptionPane.showMessageDialog(null,"Exception occurred while deleting record "+ex);
163     }
164 }
165 if(ae.getSource()==b2){
166     f.setVisible(false);
167     details d=new details();
168 }
169 if(ae.getSource()==b3){
170     f.setVisible(false);
171     details d=new details();
172 }
173 }
174
175 public static void main(String[]ar){
176     new Remove_Employee();
177 }
178
179 }
```

Output:

The screenshot displays a software window titled "Remove Employee". Inside, there is a form with the following fields and values:

- Employee Id:** 2
- Name:** Santosh Pal
- Mobile No:** 9330968220
- Email Id:** santosh21.sp4@gmail.com

Below the form, there are two buttons: "Remove" and "Cancel". A "Message" dialog box is open in the foreground, displaying an information icon and the text "deleted successfully", with an "OK" button at the bottom.

2) TESTING:

Database:

The screenshot shows a database management tool interface. On the left, the 'Navigator' pane displays a tree view of schemas, including 'javaproject' with sub-items like 'Tables', 'Views', 'Stored Procedures', 'Functions', 'samsung', and 'sys'. The main area is titled 'SQL File 5*' and contains a SQL query: `select * from login;`. Below the query, the 'Result Grid' displays the following data:

username	password
santosh	123456
dipika	0

On the right side of the interface, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'.

The screenshot shows the same database management tool interface. The 'Navigator' pane is the same. The main area is titled 'SQL File 5*' and contains a SQL query: `desc employee;`. Below the query, the 'Result Grid' displays the following data:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	NULL	
name	varchar(20)	NO		NULL	
fname	varchar(20)	NO		NULL	
agr	int	NO		NULL	
dob	varchar(20)	NO		NULL	
address	varchar(40)	NO		NULL	
phone	bigint	NO		NULL	
email	varchar(30)	NO		NULL	
education	varchar(20)	NO		NULL	
post	varchar(20)	NO		NULL	
aadhar	bigint	NO		NULL	

At the bottom of the interface, there are tabs for 'Administration' and 'Schemas'.

UI:

1)

 Login— □ ×

Username

santosh

Password


.....

Login

Cancel



2)

 Employee Details— □ ×


Employee Details

Add


View

Remove

Update




2.1)

 Add Employee

New Employee Details

Name	<input type="text" value="Santosh Pal"/>	Father's Name	<input type="text" value="Hemanta Pal"/>
Age	<input type="text" value="24"/>	Date Of Birth	<input type="text" value="15-10-1997"/>
Address	<input type="text" value="Kalighat"/>	Phone	<input type="text" value="9330968220"/>
Email Id	<input type="text" value="santosh21.sp4@gmail.com"/>	Education	<input type="text" value="M.sc"/>
Job Post	<input type="text" value="Trainee"/>	Aadhar No	<input type="text" value="777722226666"/>
Employee Id	<input type="text" value="5"/>		

Message


 Details Successfully Inserted

2.2)



A screenshot of a Windows-style dialog box titled "View". The dialog box has a light purple background with a faint pattern of stars and a large, faint outline of a map of India. In the center, the text "Employee Id" is displayed in a large, bold, black serif font. To the right of this text is a white text input field containing the number "5". Below the input field are two blue buttons with white text: "Search" on the left and "Cancel" on the right. The dialog box has a standard Windows title bar with a minimize button, a maximize button, and a close button.

2.2.1)

 Print Data


Employee Details

Employee Id:	5
Name:	Santosh Pal
Father's Name:	Hemanta Pal
Address:	Kalighat
Mobile No:	9330968220
Email Id:	santosh21.sp4@gmail.com
Education:	M.sc
Job Post:	Trainee

Print


cancel

Message

 printed successfully

OK

2.3)



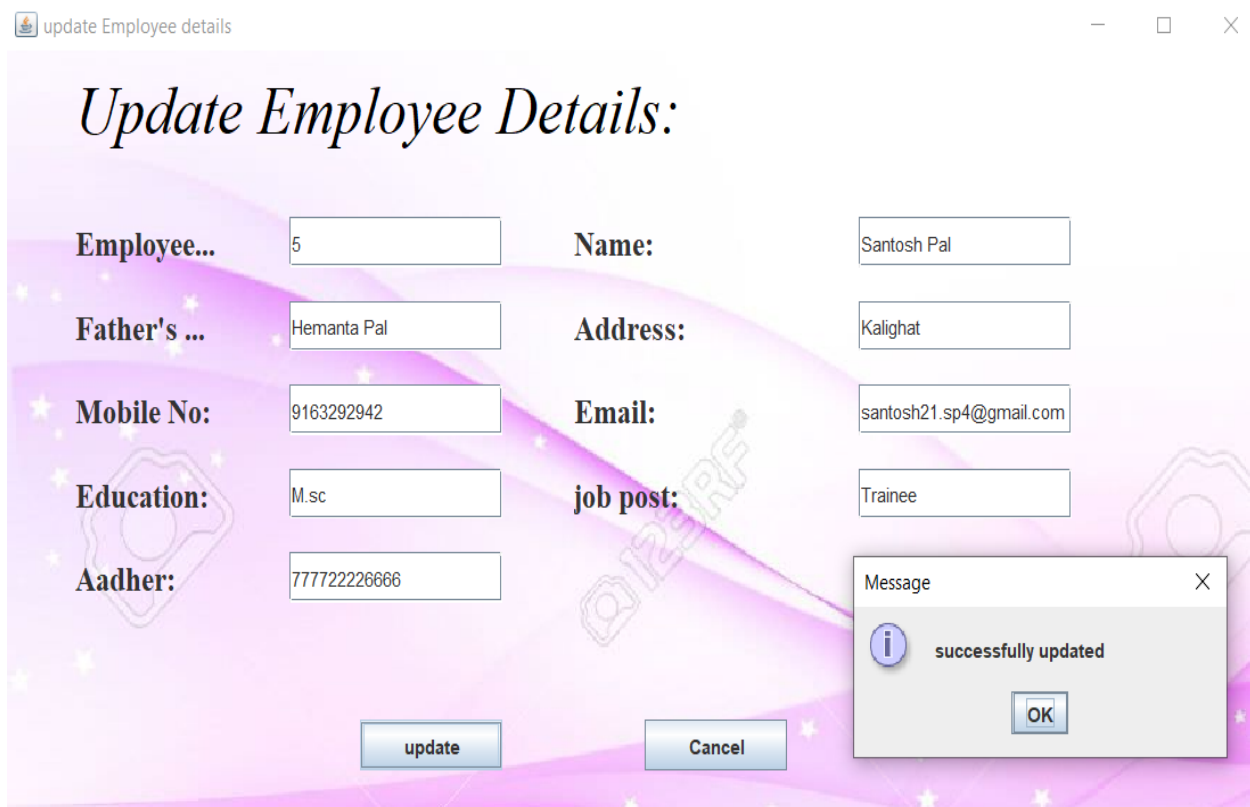
search

Employee Id

5

Search Cancel

2.3.1)



update Employee details

Update Employee Details:

Employee...	5	Name:	Santosh Pal
Father's ...	Hemanta Pal	Address:	Kalighat
Mobile No:	9163292942	Email:	santosh21.sp4@gmail.com
Education:	M.sc	job post:	Trainee
Aadher:	777722226666		

update Cancel

Message

successfully updated

OK

2.4)

The screenshot displays a software window titled "Remove Employee". Inside, there is a form with the following fields and controls:

- Employee Id:** A text box containing the value "5".
- Search:** A button located below the Employee Id field.
- back:** A button located to the right of the Search button.
- Name:** A label followed by the text "Santosh Pal".
- Mobile No:** A label followed by the text "9163292942".
- Email Id:** A label followed by the text "santosh21.sp4@gmail.com".
- Remove:** A button located below the Email Id field.
- Cancel:** A button located to the right of the Remove button.

At the bottom of the window, a "Message" dialog box is open, displaying an information icon, the text "deleted successfully", and an "OK" button.

FEATURES OF EMPLOYEE MANAGEMENT SYSTEM:

- Easy to use.
- It is completely secure.
- It is completely controlled by admin.
- This system is easily compatible with most of the web browsers.
- It is very interactive and saves time.
- Reduces paper works.
- Calculations are automated so it is highly accurate.
- Admin can view all the records whenever necessary with ease.

FUTURE SCOPE OF THE WORK:

- The option to print the records In future.
- I intend to add a leave structure in the future.
- I would like to implement a regular backup mechanism to back up the employee database to avoid disasters.
- The system can be developed in such a way that its existing features can be modified to better versions

CONCLUSION:

This project is built keeping in mind that it is to be used by only one user that is the admin. It is built for use in small scale organization where the number of employees is limited. According to the requested requirement the admin can add, manipulate, update and delete all employee data in his organization. The admin can add new departments and delete them. The Admin can also add predefined pay grades for the employees. The required records can be easily viewed by the admin anytime time he wants in an instant. The payment of the employee is based on monthly basis. Numerous validations implemented would enable the admin to enter accurate data. The main objective of this framework is to save time, make the system cost effective and management records efficiently.

REFERENCE/BIBLIOGRAPHY:

Websites:

- www.w3schools.com
- www.tutorialspoint.com
- www.youtube.com