

# Bank Management System

II B.Tech II Semester

A641X

Java Programming Internal Lab Project

Submitted By:

23211A05L0	Musku Druvitha
23211A05Q5	Puvvala Deepak
23211A05M0	Nandyala Neeraj Lochan Reddy
23211A05N4	Patlolla Harshitha Reddy



*Under the kind guidance of*

**Dr. T. Subba Reddy**

**Department of Computer Science and Engineering  
B V Raju Institute of Technology**

(UGC Autonomous, Accredited by NBA and NAAC)  
Vishnupur, Narsapur, Medak (Dist), Telangana State, India - 502313.



# Bank Management System

A Java-based GUI Banking Simulation

## 1. Introduction

The **Bank Management System** is a Java-based desktop application designed to simulate essential banking operations in a secure, user-friendly, and interactive environment. Developed using Java Swing for the graphical user interface and MySQL for backend data storage, this system mimics real-world ATM functionalities.

The project facilitates common banking features such as deposits, withdrawals, PIN changes, balance inquiries, and mini statements. All user data and transaction history are securely maintained through JDBC-based database interactions. This system is an ideal project for understanding GUI design, database connectivity, and modular programming in Java.

## 2. Modules and Methods Used

### Main Modules

- **Login**  
Authenticates the user using credentials or PIN input.
- **Transaction**  
The central interface where users choose their desired banking operations.
- **Deposit**  
Allows users to deposit a specific amount into their account.
- **Withdrawal**  
Enables users to withdraw cash after verifying sufficient balance.
- **Fast Cash**  
Offers predefined quick withdrawal options like 1000, 2000, etc.
- **Mini Statement**  
Displays a list of the most recent transactions made by the user.
- **Pin Change**  
Allows secure update of the account PIN after validation.
- **Balance Enquiry**  
Shows the current available balance in the user's account.



- **Database Connection (conn.java)**

Manages connectivity between the Java application and MySQL database.

## Key Classes and Their Methods

### Transaction.java

- Constructor initializes GUI with operation buttons.
- `actionPerformed(ActionEvent ae)`  
Detects the selected operation and launches the corresponding module.

### Deposit.java

- Accepts deposit amount input.
- Updates balance in the database.
- Records the transaction in the bank table.

### Withdrawal.java

- Takes input for withdrawal amount.
- Validates against current balance.
- Updates balance and stores the transaction.

### FastCash.java

- Displays buttons with predefined amounts.
- Deducts amount directly from balance upon selection.

### MiniStatement.java

- Queries the database for recent transactions.
- Displays the history in a readable format.

### Pin.java

- Verifies old PIN and updates to a new PIN.
- Ensures PIN format and database update.

### conn.java

- Uses JDBC to establish connection to MySQL.
- Loads JDBC driver and initializes `Connection` and `Statement` objects.



## Technologies Used

- **Java (JDK 8+)** – Core language for logic and GUI via Swing.
- **Java Swing** – For developing the graphical user interface.
- **MySQL** – Backend database for storing account and transaction data.
- **JDBC (Java Database Connectivity)** – To connect and query the MySQL database from Java.

## 3. Source Code

Below are the main Java files used in the development of the Bank Management System.

### login.java

```
1
2 package bankmanagementsystem;
3
4 import java.awt.*;
5 import java.awt.event.*;
6 import javax.swing.*;
7 import java.sql.*;
8
9 public class Login extends JFrame implements ActionListener {
10     JLabel l1, l2, l3;
11     JTextField tf1;
12     JPasswordField pf2;
13     JButton b1, b2, b3;
14
15     Login(){
16
17         setFont(new Font("System", Font.BOLD, 22));
18         Font f = getFont();
19         FontMetrics fm = getFontMetrics(f);
20         int x = fm.stringWidth("Welcome To The Bank");
21         int y = fm.stringWidth(" ");
22         int z = getWidth() - x;
23         int w = z/y;
24         String pad = "";
25         pad = String.format("%" + w + "s", pad);
26         setTitle(pad + "The Bank");
27
28
29         l1 = new JLabel("WELCOME TO BANK");
30         l1.setFont(new Font("Oswald", Font.BOLD, 38));
31
32         l2 = new JLabel("Account No:");
33         l2.setFont(new Font("Raleway", Font.BOLD, 28));
34
35         l3 = new JLabel("PIN:");
36         l3.setFont(new Font("Raleway", Font.BOLD, 28));
37
38         tf1 = new JTextField(15);
```



```

39     pf2 = new JPasswordField(15);
40
41     b1 = new JButton("SIGN IN");
42     b1.setBackground(Color.BLACK);
43     b1.setForeground(Color.WHITE);
44
45     b2 = new JButton("CLEAR");
46     b2.setBackground(Color.BLACK);
47     b2.setForeground(Color.WHITE);
48
49     b3 = new JButton("SIGN UP");
50     b3.setBackground(Color.BLACK);
51     b3.setForeground(Color.WHITE);
52
53     // Set layout and add components to frame
54     setLayout(null);
55
56     l1.setBounds(175, 50, 450, 200);
57     add(l1);
58
59     l2.setBounds(125, 150, 375, 200);
60     add(l2);
61
62     tf1.setBounds(300, 235, 230, 30);
63     add(tf1);
64
65     l3.setBounds(230, 225, 375, 200);
66     add(l3);
67
68     tf1.setFont(new Font("Arial", Font.BOLD, 14));
69
70     pf2.setFont(new Font("Arial", Font.BOLD, 14));
71     pf2.setBounds(300, 310, 230, 30);
72     add(pf2);
73
74     b1.setFont(new Font("Arial", Font.BOLD, 14));
75     b1.setBounds(130, 400, 120, 50);
76     add(b1);
77
78     b2.setFont(new Font("Arial", Font.BOLD, 14));
79     b2.setBounds(420, 400, 120, 50);
80     add(b2);
81
82     b3.setFont(new Font("Arial", Font.BOLD, 14));
83     b3.setBounds(220, 500, 230, 50);
84     add(b3);
85
86
87     b1.addActionListener(this);
88     b2.addActionListener(this);
89     b3.addActionListener(this);
90
91
92     getContentPane().setBackground(Color.gray);
93
94
95     setSize(750, 750);
96     setLocationRelativeTo(null);

```



```

97         setVisible(true);
98     }
99
100     public void actionPerformed(ActionEvent ae) {
101         try {
102             conn c1 = new conn();
103             String a = tf1.getText();
104             String b = new String(pf2.getPassword());
105
106             if (ae.getSource() == b1) {
107                 String q = "select * from login where accno = '" + a +
108                 "' and pin = '" + b + "'";
109                 ResultSet rs = c1.s.executeQuery(q);
110                 if (rs.next()) {
111                     new Transaction().setVisible(true);
112                     setVisible(false);
113                 } else {
114                     JOptionPane.showMessageDialog(null, "Incorrect
115                     Account Number or Password");
116                 }
117             } else if (ae.getSource() == b2) {
118                 tf1.setText("");
119                 pf2.setText("");
120             } else if (ae.getSource() == b3) {
121                 new Signup().setVisible(true);
122                 setVisible(false);
123             }
124         } catch (Exception e) {
125             e.printStackTrace();
126             System.out.println("error: " + e);
127         }
128     }
129
130     public static void main(String[] args) {
131         new Login().setVisible(true);
132     }
133 }

```

## Transaction.java

```

1
2 package bankmanagementsystem;
3
4 import java.awt.*;
5 import java.awt.event.*;
6 import javax.swing.*;
7 import java.sql.*;
8
9 public class Transaction extends JFrame implements ActionListener {
10
11     JLabel l1;
12     JButton b1, b2, b3, b4, b5, b6, b7;
13
14     Transaction() {
15
16         setFont(new Font("System", Font.BOLD, 22));
17         Font f = getFont();
18         FontMetrics fm = getFontMetrics(f);

```



```

19     int x = fm.stringWidth("TRANSACTION");
20     int y = fm.stringWidth(" ");
21     int z = getWidth() - (3 * x);
22     int w = z / y;
23     String pad = "";
24
25     pad = String.format("%" + w + "s", pad);
26     setTitle(pad + "TRANSACTION");
27
28     l1 = new JLabel("Please Select Your Transaction");
29     l1.setFont(new Font("System", Font.BOLD, 38));
30
31     b1 = new JButton("DEPOSIT");
32     b1.setFont(new Font("System", Font.BOLD, 18));
33     b1.setBackground(Color.BLACK);
34     b1.setForeground(Color.WHITE);
35
36     b2 = new JButton("CASH WITHDRAWAL");
37     b2.setFont(new Font("System", Font.BOLD, 18));
38     b2.setBackground(Color.BLACK);
39     b2.setForeground(Color.WHITE);
40
41     b3 = new JButton("FAST CASH");
42     b3.setFont(new Font("System", Font.BOLD, 18));
43     b3.setBackground(Color.BLACK);
44     b3.setForeground(Color.WHITE);
45
46     b4 = new JButton("MINI STATEMENT");
47     b4.setFont(new Font("System", Font.BOLD, 18));
48     b4.setBackground(Color.BLACK);
49     b4.setForeground(Color.WHITE);
50
51     b5 = new JButton("PIN CHANGE");
52     b5.setFont(new Font("System", Font.BOLD, 18));
53     b5.setBackground(Color.BLACK);
54     b5.setForeground(Color.WHITE);
55
56     b6 = new JButton("BALANCE ENQUIRY");
57     b6.setFont(new Font("System", Font.BOLD, 18));
58     b6.setBackground(Color.BLACK);
59     b6.setForeground(Color.WHITE);
60
61     b7 = new JButton("EXIT");
62     b7.setFont(new Font("System", Font.BOLD, 18));
63     b7.setBackground(Color.BLACK);
64     b7.setForeground(Color.WHITE);
65
66     setLayout(null);
67
68     l1.setBounds(100, 100, 700, 40);
69     add(l1);
70
71     b1.setBounds(40, 250, 300, 60);
72     add(b1);
73
74     b2.setBounds(440, 250, 300, 60);
75     add(b2);
76

```



```

77     b3.setBounds(40, 360, 300, 60);
78     add(b3);
79
80     b4.setBounds(440, 360, 300, 60);
81     add(b4);
82
83     b5.setBounds(40, 470, 300, 60);
84     add(b5);
85
86     b6.setBounds(440, 470, 300, 60);
87     add(b6);
88
89     b7.setBounds(240, 600, 300, 60);
90     add(b7);
91
92     b1.addActionListener(this);
93     b2.addActionListener(this);
94     b3.addActionListener(this);
95     b4.addActionListener(this);
96     b5.addActionListener(this);
97     b6.addActionListener(this);
98     b7.addActionListener(this);
99
100    getContentPane().setBackground(Color.gray);
101
102    setSize(800, 800);
103    setLocationRelativeTo(null); // Center the frame
104    setVisible(true);
105
106 }
107
108 public void actionPerformed(ActionEvent ae) {
109
110     if (ae.getSource() == b1) {
111
112         new Deposit().setVisible(true);
113         setVisible(false);
114
115     } else if (ae.getSource() == b2) {
116
117         new Withdrawl().setVisible(true);
118         setVisible(false);
119
120     } else if (ae.getSource() == b3) {
121
122         new FastCash().setVisible(true);
123         setVisible(false);
124
125     } else if (ae.getSource() == b4) {
126
127         new MiniStatement().setVisible(true);
128         setVisible(false);
129
130     } else if (ae.getSource() == b5) {
131
132         new Pin().setVisible(true);
133         setVisible(false);
134

```



```

135     } else if (ae.getSource() == b6) {
136
137         String pinn = JOptionPane.showInputDialog("Enter PIN");
138         conn c1 = new conn();
139
140         try {
141
142             ResultSet rs = c1.s.executeQuery(" SELECT balance FROM
bank where pin =' " + pinn + "' ORDER BY sno DESC LIMIT 1");
143
144             if (rs.next()) {
145
146                 String balance = rs.getString("balance");
147
148                 JOptionPane.showMessageDialog(null, "Your Account
Balance is " + balance);
149
150             }
151
152             } catch (Exception e) {
153
154                 e.printStackTrace();
155
156             }
157
158     } else if (ae.getSource() == b7) {
159
160         System.exit(0);
161
162     }
163
164 }
165
166 public static void main(String[] args) {
167     new Transaction().setVisible(true);
168 }
169 }

```

## deposit.java

```

1
2 package bankmanagementsystem;
3
4 import java.awt.*;
5 import java.awt.event.*;
6 import javax.swing.*;
7 import java.sql.*;
8
9 public class Deposit extends JFrame implements ActionListener {
10
11     JTextField t1, t2;
12     JButton b1, b2, b3;
13     JLabel l1, l2, l3;
14
15     // Maximum deposit limit
16     private static final double DEPOSIT_LIMIT = 500000.00;
17     private boolean pinValidated = false; // Track if the PIN is
validated

```



```

18
19 Deposit() {
20     setFont(new Font("System", Font.BOLD, 22));
21     Font f = getFont();
22     FontMetrics fm = getFontMetrics(f);
23     int x = fm.stringWidth("DEPOSIT");
24     int y = fm.stringWidth(" ");
25     int z = getWidth() - (5 * x);
26     int w = z / y;
27     String pad = "";
28
29     pad = String.format("%" + w + "s", pad);
30     setTitle(pad + "DEPOSIT");
31
32     l1 = new JLabel("ENTER AMOUNT YOU WANT");
33     l1.setFont(new Font("System", Font.BOLD, 35));
34
35     l2 = new JLabel("TO DEPOSIT");
36     l2.setFont(new Font("System", Font.BOLD, 35));
37
38     l3 = new JLabel("Enter Pin");
39     l3.setFont(new Font("System", Font.BOLD, 14));
40
41     t1 = new JTextField();
42     t1.setFont(new Font("Raleway", Font.BOLD, 22));
43
44     t2 = new JTextField();
45     t2.setFont(new Font("Raleway", Font.BOLD, 14));
46
47     b1 = new JButton("ENTER PIN");
48     b1.setFont(new Font("System", Font.BOLD, 18));
49     b1.setBackground(Color.BLACK);
50     b1.setForeground(Color.WHITE);
51
52     b2 = new JButton("BACK");
53     b2.setFont(new Font("System", Font.BOLD, 18));
54     b2.setBackground(Color.BLACK);
55     b2.setForeground(Color.WHITE);
56
57     b3 = new JButton("EXIT");
58     b3.setFont(new Font("System", Font.BOLD, 18));
59     b3.setBackground(Color.BLACK);
60     b3.setForeground(Color.WHITE);
61
62     setLayout(null);
63
64     l3.setBounds(620, 10, 80, 30);
65     add(l3);
66
67     t2.setBounds(700, 10, 40, 30);
68     add(t2);
69
70     l1.setBounds(150, 150, 800, 60);
71     add(l1);
72
73     l2.setBounds(290, 210, 800, 60);
74     add(l2);
75

```



```

76     t1.setBounds(250, 300, 300, 50);
77     add(t1);
78
79     b1.setBounds(180, 380, 170, 50);
80     add(b1);
81
82     b2.setBounds(445, 380, 125, 50);
83     add(b2);
84
85     b3.setBounds(300, 500, 200, 50);
86     add(b3);
87
88     b1.addActionListener(this);
89     b2.addActionListener(this);
90     b3.addActionListener(this);
91
92     // Input validation to allow only digits in the deposit field
93     t1.addKeyListener(new KeyAdapter() {
94         public void keyTyped(KeyEvent ke) {
95             char c = ke.getKeyChar();
96             // Allow only digits, backspace, and delete
97             if (!Character.isDigit(c) && c != KeyEvent.
VK_BACK_SPACE) {
98                 ke.consume(); // Ignore non-digit characters
99                 JOptionPane.showMessageDialog(null, "Please enter
digits only.");
100             }
101         }
102     });
103
104     // Initially, disable the deposit field
105     t1.setEnabled(false);
106
107     getContentPane().setBackground(Color.gray);
108
109     setSize(800, 800);
110     setLocationRelativeTo(null); // Center the frame
111     setVisible(true);
112 }
113
114 public void actionPerformed(ActionEvent ae) {
115     try {
116         String a = t1.getText();
117         String b = t2.getText();
118
119         if (ae.getSource() == b1) {
120             if (!pinValidated) {
121                 // Step 1: Validate the PIN entered
122                 if (b.equals("")) {
123                     JOptionPane.showMessageDialog(null, "Please
enter your PIN.");
124                     return;
125                 }
126
127                 conn c1 = new conn();
128
129                 // Query the database to get user details using the
entered PIN

```



```

130         ResultSet rs = c1.s.executeQuery("select * from
bank where pin = '" + b + "'");
131
132         if (rs.next()) {
133             pinValidated = true; // Pin is validated
134             JOptionPane.showMessageDialog(null, "PIN
validated. Please enter the deposit amount.");
135
136             // Enable the deposit amount field
137             t1.setEnabled(true);
138             t1.setText(""); // Clear the deposit amount
field
139
140             // Change the button text to "DEPOSIT"
141             b1.setText("DEPOSIT");
142             b1.removeActionListener(this);
143             b1.addActionListener(this); // Rebind action
listener for deposit
144         } else {
145             JOptionPane.showMessageDialog(null, "Invalid
PIN! Please try again.");
146             t2.setText(""); // Clear the PIN field
147         }
148     } else {
149         // Step 2: After PIN validation, proceed with
deposit
150
151         // Check if deposit amount is entered
152         if (a.equals("")) {
153             JOptionPane.showMessageDialog(null, "Please
enter the Amount you want to Deposit");
154             return;
155         }
156
157         // Convert deposit amount to double
158         double depositAmount = Double.parseDouble(a);
159
160         // Check if deposit amount exceeds the limit
161         if (depositAmount > DEPOSIT_LIMIT) {
162             JOptionPane.showMessageDialog(null, "Deposit
limit exceeded! The maximum deposit amount is Rs. " + DEPOSIT_LIMIT);
;
163             return;
164         }
165
166         conn c1 = new conn();
167
168         // Query the database to get user details using the
entered PIN
169         ResultSet rs = c1.s.executeQuery("select * from
bank where pin = '" + b + "' order by sno desc limit 1");
170
171         if (rs.next()) {
172             String pin = rs.getString("pin");
173             String accno = rs.getString("accno");
174             double sno = rs.getDouble("sno");
175             double bal = rs.getDouble("balance");
176             System.out.println(bal);

```



```

177         // Update balance
178         bal = bal + depositAmount;
179         sno++;
180
181         // Insert new transaction record
182         String q1 = "insert into bank(pin,deposit,
183         withdraw,balance,accno,sno) values('" + pin + "','"+ depositAmount
184         + "','0,'"+ bal + "','"+ accno + "','"+ sno + "')";
185         c1.s.executeUpdate(q1);
186     }
187
188     // Notify the user about the successful deposit
189     JOptionPane.showMessageDialog(null, "Rs. " + a + "
190     Deposited Successfully");
191     new Transaction().setVisible(true);
192     setVisible(false);
193 }
194
195 } else if (ae.getSource() == b2) {
196     // Back button action
197     new Transaction().setVisible(true);
198     setVisible(false);
199 }
200
201 } else if (ae.getSource() == b3) {
202     // Exit button action
203     System.exit(0);
204 }
205
206 } catch (Exception e) {
207     e.printStackTrace();
208     System.out.println("Error: " + e);
209 }
210
211 }
212
213 public static void main(String[] args) {
214     new Deposit().setVisible(true);
215 }
216
217 }

```

## withdrawl.java

```

1
2 package bankmanagementsystem;
3
4 import java.awt.*;
5 import java.awt.event.*;
6 import javax.swing.*;
7 import java.sql.*;
8
9 public class Withdrawl extends JFrame implements ActionListener {
10
11     JTextField t1, t2;
12     JButton b1, b2, b3;
13     JLabel l1, l2, l3, l4;
14
15     // Withdrawal limit
16     private static final double WITHDRAWAL_LIMIT = 10000.00;
17     private boolean pinValidated = false; // Track if the PIN is

```



validated

```
18
19 Withdrawl() {
20     setFont(new Font("System", Font.BOLD, 22));
21     Font f = getFont();
22     FontMetrics fm = getFontMetrics(f);
23     int x = fm.stringWidth("WITHDRAWAL");
24     int y = fm.stringWidth(" ");
25     int z = getWidth() - (3 * x);
26     int w = z / y;
27     String pad = "";
28     pad = String.format("%" + w + "s", pad);
29     setTitle(pad + "WITHDRAWAL");
30
31     l1 = new JLabel("MAXIMUM DAILY WITHDRAWAL");
32     l1.setFont(new Font("System", Font.BOLD, 40));
33
34     l2 = new JLabel("IS RS.10,000");
35     l2.setFont(new Font("System", Font.BOLD, 40));
36
37     l3 = new JLabel("PLEASE ENTER YOUR AMOUNT");
38     l3.setFont(new Font("System", Font.BOLD, 35));
39
40     l4 = new JLabel("Enter Pin");
41     l4.setFont(new Font("System", Font.BOLD, 14));
42
43     t1 = new JTextField();
44     t1.setFont(new Font("Raleway", Font.BOLD, 22));
45
46     t2 = new JTextField();
47     t2.setFont(new Font("Raleway", Font.BOLD, 14));
48
49     b1 = new JButton("ENTER PIN");
50     b1.setFont(new Font("System", Font.BOLD, 18));
51     b1.setBackground(Color.BLACK);
52     b1.setForeground(Color.WHITE);
53
54     b2 = new JButton("BACK");
55     b2.setFont(new Font("System", Font.BOLD, 18));
56     b2.setBackground(Color.BLACK);
57     b2.setForeground(Color.WHITE);
58
59     b3 = new JButton("EXIT");
60     b3.setFont(new Font("System", Font.BOLD, 18));
61     b3.setBackground(Color.BLACK);
62     b3.setForeground(Color.WHITE);
63
64     setLayout(null);
65
66     l4.setBounds(620, 10, 80, 30);
67     add(l4);
68
69     t2.setBounds(700, 10, 40, 30);
70     add(t2);
71
72     l1.setBounds(90, 100, 800, 60);
73     add(l1);
74
```



```

75     t2.setBounds(270, 160, 800, 60);
76     add(t2);
77
78     t3.setBounds(120, 260, 800, 60);
79     add(t3);
80
81     t1.setBounds(210, 340, 360, 50);
82     add(t1);
83
84     b1.setBounds(220, 400, 160, 50);
85     add(b1);
86
87     b2.setBounds(400, 400, 160, 50);
88     add(b2);
89
90     b3.setBounds(300, 550, 200, 50);
91     add(b3);
92
93     b1.addActionListener(this);
94     b2.addActionListener(this);
95     b3.addActionListener(this);
96
97     // Input validation to allow only digits in the withdrawal
field
98     t1.addKeyListener(new KeyAdapter() {
99         public void keyTyped(KeyEvent ke) {
100             char c = ke.getKeyChar();
101             // Allow only digits and backspace
102             if (!Character.isDigit(c) && c != KeyEvent.
VK_BACK_SPACE) {
103                 ke.consume(); // Ignore non-digit characters
104                 JOptionPane.showMessageDialog(null, "Please enter
digits only.");
105             }
106         }
107     });
108
109     // Initially, disable the withdrawal field
110     t1.setEnabled(false);
111
112     getContentPane().setBackground(Color.gray);
113     setSize(800, 800);
114     setLocationRelativeTo(null); // Center the frame
115     setVisible(true);
116 }
117
118 public void actionPerformed(ActionEvent ae) {
119     try {
120         String a = t1.getText();
121         String b = t2.getText();
122
123         if (ae.getSource() == b1) {
124             if (!pinValidated) {
125                 // Step 1: Validate the PIN entered
126                 if (b.equals("")) {
127                     JOptionPane.showMessageDialog(null, "Please
enter your PIN.");
128                     return;

```



```

129         }
130
131         conn c1 = new conn();
132
133         // Query the database to get user details using the
134         entered PIN
135         ResultSet rs = c1.s.executeQuery("select * from
136         bank where pin = '" + b + "'");
137
138         if (rs.next()) {
139             pinValidated = true; // Pin is validated
140             JOptionPane.showMessageDialog(null, "PIN
141             validated. Please enter the withdrawal amount.");
142
143             // Enable the withdrawal amount field
144             t1.setEnabled(true);
145             t1.setText(""); // Clear the withdrawal amount
146             field
147
148             // Change the button text to "WITHDRAW"
149             b1.setText("WITHDRAW");
150             b1.removeActionListener(this);
151             b1.addActionListener(this); // Rebind action
152             listener for withdrawal
153         } else {
154             JOptionPane.showMessageDialog(null, "Invalid
155             PIN! Please try again.");
156             t2.setText(""); // Clear the PIN field
157         }
158     } else {
159         // Step 2: After PIN validation, proceed with
160         withdrawal
161
162         // Check if withdrawal amount is entered
163         if (a.equals("")) {
164             JOptionPane.showMessageDialog(null, "Please
165             enter the Amount you want to Withdraw");
166             return;
167         }
168
169         // Convert withdrawal amount to double
170         double withdrawalAmount = Double.parseDouble(a);
171
172         // Check if withdrawal amount exceeds the limit
173         if (withdrawalAmount > WITHDRAWAL_LIMIT) {
174             JOptionPane.showMessageDialog(null, "Withdrawal
175             limit exceeded! The maximum withdrawal amount is Rs. " +
176             WITHDRAWAL_LIMIT);
177             return;
178         }
179
180         conn c1 = new conn();
181
182         // Query the database to get user details using the
183         entered PIN
184         ResultSet rs = c1.s.executeQuery("select * from
185         bank where pin = '" + b + "' order by sno desc limit 1");

```



```

175         double balance = 0;
176         if (rs.next()) {
177             String pin = rs.getString("pin");
178             String accno = rs.getString("accno");
179
180             balance = rs.getDouble("balance");
181             double sno = rs.getDouble("sno");
182             sno++;
183
184             // Check if the balance is sufficient for
withdrawal
185             if (balance >= withdrawalAmount) {
186                 balance -= withdrawalAmount; // Deduct the
withdrawal amount
187                 String q1 = "insert into bank(pin, deposit,
withdraw, balance, accno, sno) values('" + pin + "', 0, '" +
withdrawalAmount + "', '" + balance + "', '" + accno + "', '" + sno
+ "');"
188                 c1.s.executeUpdate(q1);
189             } else {
190                 JOptionPane.showMessageDialog(null, "
Insufficient Balance to Withdraw Rs. " + withdrawalAmount);
191                 return;
192             }
193         }
194
195         // Notify the user about the successful withdrawal
196         JOptionPane.showMessageDialog(null, "Rs. " + a + "
Debited Successfully");
197
198         new Transaction().setVisible(true);
199         setVisible(false);
200     }
201
202     } else if (ae.getSource() == b2) {
203         // Back button action
204         new Transaction().setVisible(true);
205         setVisible(false);
206
207     } else if (ae.getSource() == b3) {
208         // Exit button action
209         System.exit(0);
210     }
211     } catch (Exception e) {
212         e.printStackTrace();
213         System.out.println("Error: " + e);
214     }
215 }
216
217 public static void main(String[] args) {
218     new Withdrawl().setVisible(true);
219 }
220 }

```

## fastcash.java

```

1
2 package bankmanagementsystem;

```



```

3
4
5
6
7 import java.awt.*;
8 import java.awt.event.*;
9 import javax.swing.*;
10 import java.sql.*;
11 import java.util.*;
12
13
14 public class FastCash extends JFrame implements ActionListener{
15
16     JLabel l1,l2;
17     JButton b1,b2,b3,b4,b5,b6,b7,b8;
18     JTextField t1;
19
20     FastCash(){
21
22         setFont(new Font("System", Font.BOLD, 22));
23         Font f = getFont();
24         FontMetrics fm = getFontMetrics(f);
25         int x = fm.stringWidth("FAST CASH");
26         int y = fm.stringWidth(" ");
27         int z = getWidth() - (4*x);
28         int w = z/y;
29         String pad = "";
30         //for (int i=0; i!=w; i++) pad += " ";
31         pad = String.format("%"+w+"s", pad);
32         setTitle(pad+"FAST CASH");
33
34
35
36
37         l1 = new JLabel("SELECT WITHDRAWAL AMOUNT");
38         l1.setFont(new Font("System", Font.BOLD, 38));
39
40         l2 = new JLabel("Enter PIN");
41         l2.setFont(new Font("System", Font.BOLD, 13));
42
43         t1 = new JTextField();
44         t1.setFont(new Font("System", Font.BOLD, 13));
45
46
47         b1 = new JButton("Rs 100");
48         b1.setFont(new Font("System", Font.BOLD, 18));
49         b1.setBackground(Color.BLACK);
50         b1.setForeground(Color.WHITE);
51
52         b2 = new JButton("Rs 500");
53         b2.setFont(new Font("System", Font.BOLD, 18));
54         b2.setBackground(Color.BLACK);
55         b2.setForeground(Color.WHITE);
56
57         b3 = new JButton("Rs 1000");
58         b3.setFont(new Font("System", Font.BOLD, 18));
59         b3.setBackground(Color.BLACK);
60         b3.setForeground(Color.WHITE);

```



```

61
62     b4 = new JButton("Rs 2000");
63     b4.setFont(new Font("System", Font.BOLD, 18));
64     b4.setBackground(Color.BLACK);
65     b4.setForeground(Color.WHITE);
66
67     b5 = new JButton("Rs 5000");
68     b5.setFont(new Font("System", Font.BOLD, 18));
69     b5.setBackground(Color.BLACK);
70     b5.setForeground(Color.WHITE);
71
72     b6 = new JButton("Rs 10000");
73     b6.setFont(new Font("System", Font.BOLD, 18));
74     b6.setBackground(Color.BLACK);
75     b6.setForeground(Color.WHITE);
76
77     b7 = new JButton("BACK");
78     b7.setFont(new Font("System", Font.BOLD, 18));
79     b7.setBackground(Color.BLACK);
80     b7.setForeground(Color.WHITE);
81
82     b7 = new JButton("EXIT");
83     b7.setFont(new Font("System", Font.BOLD, 18));
84     b7.setBackground(Color.BLACK);
85     b7.setForeground(Color.WHITE);
86
87
88     setLayout(null);
89
90     l2.setBounds(640,10,60,40);
91     add(l2);
92
93     t1.setBounds(710,10,60,40);
94     add(t1);
95
96     l1.setBounds(100,100,700,40);
97     add(l1);
98
99     b1.setBounds(40,250,300,60);
100    add(b1);
101
102    b2.setBounds(440,250,300,60);
103    add(b2);
104
105    b3.setBounds(40,360,300,60);
106    add(b3);
107
108    b4.setBounds(440,360,300,60);
109    add(b4);
110
111    b5.setBounds(40,470,300,60);
112    add(b5);
113
114    b6.setBounds(440,470,300,60);
115    add(b6);
116
117    b7.setBounds(240,600,300,60);
118    add(b7);

```



```

119
120
121     b1.addActionListener(this);
122     b2.addActionListener(this);
123     b3.addActionListener(this);
124     b4.addActionListener(this);
125     b5.addActionListener(this);
126     b6.addActionListener(this);
127     b7.addActionListener(this);
128     getContentPane().setBackground(Color.gray);
129
130
131
132
133     setSize(800,800);
134     setLocationRelativeTo(null);
135     setVisible(true);
136
137
138
139 }
140
141 public void actionPerformed(ActionEvent ae){
142
143     try{
144
145         String a = t1.getText();
146         double balance = 0;
147
148         if(ae.getSource()==b1){
149
150             conn c1 = new conn();
151
152             ResultSet rs = c1.s.executeQuery(" select * from
bank where pin = '"+a+"' order by sno desc limit 1");
153
154
155             if(rs.next()){
156                 String pin = rs.getString("pin");
157                 String accno = rs.getString("accno");
158
159                 balance = rs.getDouble("balance");
160                 double sno = rs.getDouble("sno");
161                 sno++;
162
163                 balance -=100;
164                 String q1= "insert into bank values('"+pin+"',
null,100,'"+balance+"','"+accno+"','"+sno+"')";
165
166                 c1.s.executeUpdate(q1);
167             }
168
169
170
171             JOptionPane.showMessageDialog(null, "Rs. "+100+"
Debited Successfully");
172
173             new Transaction().setVisible(true);

```



```

174         setVisible(false);
175
176
177
178     }
179
180
181     else if(ae.getSource()==b2){
182
183         conn c1 = new conn();
184
185         ResultSet rs = c1.s.executeQuery(" select * from
bank where pin = '"+a+"' order by sno desc limit 1 order by sno desc
limit 1 ");
186
187
188         if(rs.next()){
189             String pin = rs.getString("pin");
190             String accno = rs.getString("accno");
191             balance = rs.getDouble("balance");
192             double sno = rs.getDouble("sno");
193             sno++;
194             balance-=500;
195             String q1= "insert into bank values('"+pin+"',
null,500,'"+balance+"','"+accno+"','"+sno+"')";
196
197             c1.s.executeUpdate(q1);
198         }
199
200
201
202         JOptionPane.showMessageDialog(null, "Rs. "+500+"
Debited Successfully");
203
204         new Transaction().setVisible(true);
205         setVisible(false);
206
207
208
209     }
210     else if(ae.getSource()==b3){
211
212         conn c1 = new conn();
213
214         ResultSet rs = c1.s.executeQuery(" select * from
bank where pin = '"+a+"' order by sno desc limit 1 ");
215
216
217         if(rs.next()){
218             String pin = rs.getString("pin");
219             String accno = rs.getString("accno");
220             balance = rs.getDouble("balance");
221             double sno = rs.getDouble("sno");
222             sno++;
223             balance-=1000;
224             String q1= "insert into bank values('"+pin+"',
null,1000,'"+balance+"','"+accno+"','"+sno+"')";
225

```



```

226         c1.s.executeUpdate(q1);
227     }
228
229
230
231     JOptionPane.showMessageDialog(null, "Rs. "+1000+"
Debited Successfully");
232
233     new Transaction().setVisible(true);
234     setVisible(false);
235
236
237
238 }
239 else if(ae.getSource()==b4){
240
241     conn c1 = new conn();
242
243     ResultSet rs = c1.s.executeQuery(" select * from
bank where pin = '"+a+"' order by sno desc limit 1 ");
244
245
246     if(rs.next()){
247         String pin = rs.getString("pin");
248         String accno = rs.getString("accno");
249         balance = rs.getDouble("balance");
250         double sno = rs.getDouble("sno");
251         sno++;
252         balance-=2000;
253         String q1= "insert into bank values('"+pin+"',
null,2000,'"+balance+"', '"+accno+"', '"+sno+"')";
254
255         c1.s.executeUpdate(q1);
256     }
257
258
259
260     JOptionPane.showMessageDialog(null, "Rs. "+2000+"
Debited Successfully");
261
262     new Transaction().setVisible(true);
263     setVisible(false);
264
265
266
267 }
268 else if(ae.getSource()==b5){
269
270     conn c1 = new conn();
271
272     ResultSet rs = c1.s.executeQuery(" select * from
bank where pin = '"+a+"' order by sno desc limit 1");
273
274
275     if(rs.next()){
276         String pin = rs.getString("pin");
277         String accno = rs.getString("accno");
278         balance = rs.getDouble("balance");

```



```

279         double sno = rs.getDouble("sno");
280         sno++;
281         balance -= 5000;
282         String q1= "insert into bank values('"+pin+"',
null,5000,'"+balance+"', '"+accno+"', '"+sno+"')";
283
284         c1.s.executeUpdate(q1);
285     }
286
287
288
289     JOptionPane.showMessageDialog(null, "Rs. "+5000+"
Debited Successfully");
290
291     new Transaction().setVisible(true);
292     setVisible(false);
293
294
295
296     }
297     else if(ae.getSource()==b6){
298
299         conn c1 = new conn();
300
301         ResultSet rs = c1.s.executeQuery(" select * from
bank where pin = '"+a+"'' order by sno desc limit 1 ");
302
303
304         if(rs.next()){
305             String pin = rs.getString("pin");
306             String accno = rs.getString("accno");
307             balance = rs.getDouble("balance");
308             double sno = rs.getDouble("sno");
309             sno++;
310             balance -= 10000;
311             String q1= "insert into bank values('"+pin+"',
null,10000,'"+balance+"', '"+accno+"', '"+sno+"')";
312
313             c1.s.executeUpdate(q1);
314         }
315
316
317
318         JOptionPane.showMessageDialog(null, "Rs. "+10000+"
Debited Successfully");
319
320         new Transaction().setVisible(true);
321         setVisible(false);
322
323
324
325     }
326
327
328     else if(ae.getSource()==b7){
329
330         System.exit(0);
331

```



```

332     }
333     }catch(Exception e){
334         e.printStackTrace();
335         System.out.println("error: "+e);
336     }
337
338 }
339
340
341 public static void main(String[] args){
342     new FastCash().setVisible(true);
343 }
344 }

```

## ministatement.java

```

1  /*
2   * To change this license header, choose License Headers in Project
   * Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package bankmanagementsystem;
7
8  /**
9   *
10  * @author maini_
11  */
12
13
14  import java.awt.*;
15  import java.awt.event.*;
16  import javax.swing.*;
17  import java.sql.*;
18
19  public class MiniStatement extends JFrame implements ActionListener{
20
21      JTable t1;
22      JButton b1;
23      String x[] = {"Account number","Deposit","Withdraw","Balance"};
24      String y[][] = new String[20][5];
25      int i=0, j=0;
26
27      MiniStatement(){
28          super("Mini Statement");
29          setSize(1200,650);
30          setLocationRelativeTo(null);
31          getContentPane().setBackground(Color.gray);
32          String pinn = JOptionPane.showInputDialog("Enter PIN");
33
34
35          try{
36              conn c1 = new conn();
37              ResultSet rs = c1.s.executeQuery("SELECT * FROM bank where
pin = '" + pinn + "' ");
38              // String s1 = "select * from bank";
39              // ResultSet rs = c1.s.executeQuery(s1);
40              while(rs.next()){

```



```

41         y[i][j++]=rs.getString("accno");
42         //y[i][j++]=rs.getString("date");
43         y[i][j++]=rs.getString("deposit");
44         y[i][j++]=rs.getString("withdraw");
45         y[i][j++]=rs.getString("balance");
46         i++;
47         j=0;
48     }
49     t1 = new JTable(y,x);
50
51     }catch(Exception e){
52         e.printStackTrace();
53     }
54
55
56     b1 = new JButton("Print");
57     add(b1,"South");
58     JScrollPane sp = new JScrollPane(t1);
59     add(sp);
60     b1.addActionListener(this);
61 }
62 public void actionPerformed(ActionEvent ae){
63     try{
64         t1.print();
65     }catch(Exception e){}
66 }
67
68 public static void main(String[] args){
69     new MiniStatement().setVisible(true);
70 }
71
72 }

```

## pin.java

```

1  /*
2   * To change this license header, choose License Headers in Project
   * Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package bankmanagementsystem;
7
8  import java.awt.*;
9  import java.awt.event.*;
10 import javax.swing.*;
11 import java.sql.*;
12
13 public class Pin extends JFrame implements ActionListener {
14
15     JPasswordField t1, t2, t3;
16     JButton b1, b2;
17     JLabel l1, l2, l3, l4;
18
19     Pin() {
20
21         setFont(new Font("System", Font.BOLD, 22));
22         Font f = getFont();

```



```

23     FontMetrics fm = getFontMetrics(f);
24     int x = fm.stringWidth("PIN CHANGE");
25     int y = fm.stringWidth(" ");
26     int z = getWidth() - (3 * x);
27     int w = z / y;
28     String pad = "";
29
30     pad = String.format("%" + w + "s", pad);
31     setTitle(pad + "PIN CHANGE");
32
33
34     l1 = new JLabel("CHANGE YOUR PIN");
35     l1.setFont(new Font("System", Font.BOLD, 35));
36
37     l2 = new JLabel("Current PIN:");
38     l2.setFont(new Font("System", Font.BOLD, 22));
39
40     l3 = new JLabel("New PIN:");
41     l3.setFont(new Font("System", Font.BOLD, 22));
42
43     l4 = new JLabel("Re-Enter New PIN:");
44     l4.setFont(new Font("System", Font.BOLD, 22));
45
46     t1 = new JPasswordField();
47     t1.setFont(new Font("Raleway", Font.BOLD, 22));
48
49     t2 = new JPasswordField();
50     t2.setFont(new Font("Raleway", Font.BOLD, 22));
51
52     t3 = new JPasswordField();
53     t3.setFont(new Font("Raleway", Font.BOLD, 22));
54
55     b1 = new JButton("SAVE");
56     b1.setFont(new Font("System", Font.BOLD, 18));
57     b1.setBackground(Color.BLACK);
58     b1.setForeground(Color.WHITE);
59
60     b2 = new JButton("BACK");
61     b2.setFont(new Font("System", Font.BOLD, 18));
62     b2.setBackground(Color.BLACK);
63     b2.setForeground(Color.WHITE);
64
65     b1.addActionListener(this);
66     b2.addActionListener(this);
67
68     setLayout(null);
69
70     l1.setBounds(220, 130, 800, 60);
71     add(l1);
72
73     l2.setBounds(100, 240, 150, 40);
74     add(l2);
75
76     l3.setBounds(100, 300, 150, 40);
77     add(l3);
78
79     l4.setBounds(100, 360, 200, 40);
80     add(l4);

```



```

81
82     t1.setBounds(310, 240, 360, 40);
83     add(t1);
84
85     t2.setBounds(310, 300, 360, 40);
86     add(t2);
87
88     t3.setBounds(310, 360, 360, 40);
89     add(t3);
90
91     b1.setBounds(220, 460, 160, 50);
92     add(b1);
93
94     b2.setBounds(400, 460, 160, 50);
95     add(b2);
96
97     getContentPane().setBackground(Color.gray);
98
99     setSize(800, 800);
100    setLocationRelativeTo(null);
101    setVisible(true);
102 }
103
104 public void actionPerformed(ActionEvent ae) {
105
106     try {
107         String a = new String(t1.getPassword()); // Get the current
PIN
108         String b = new String(t2.getPassword()); // Get the new PIN
109         String c = new String(t3.getPassword()); // Get the re-
entered new PIN
110
111         // Validate that the PINs are 4 digits long and numeric
112         if (ae.getSource() == b1) {
113
114             if (a.equals("") || b.equals("") || c.equals("")) {
115                 JOptionPane.showMessageDialog(null, "Please fill
all the fields.");
116                 return;
117             }
118
119             if (a.length() != 4 || !a.matches("\\d+")) {
120                 JOptionPane.showMessageDialog(null, "Current PIN
must be a 4-digit number.");
121                 return;
122             }
123
124             if (b.length() != 4 || !b.matches("\\d+")) {
125                 JOptionPane.showMessageDialog(null, "New PIN must
be a 4-digit number.");
126                 return;
127             }
128
129             if (c.length() != 4 || !c.matches("\\d+")) {
130                 JOptionPane.showMessageDialog(null, "Re-entered PIN
must be a 4-digit number.");
131                 return;
132             }

```



```

133         if (b.equals(c)) {
134             conn c1 = new conn();
135             String q1 = "update bank set pin = '" + b + "'
136 where pin = '" + a + "'";
137             String q2 = "update login set pin = '" + b + "'
138 where pin = '" + a + "'";
139             String q3 = "update signup3 set pin = '" + b + "'
140 where pin = '" + a + "'";
141
142             c1.s.executeUpdate(q1);
143             c1.s.executeUpdate(q2);
144             c1.s.executeUpdate(q3);
145
146             JOptionPane.showMessageDialog(null, "PIN changed
147 successfully");
148
149             new Transaction().setVisible(true);
150             setVisible(false);
151         } else {
152             JOptionPane.showMessageDialog(null, "PIN entered
153 doesn't match.");
154         }
155     } else if (ae.getSource() == b2) {
156         new Transaction().setVisible(true);
157         setVisible(false);
158     }
159 } catch (Exception e) {
160     e.printStackTrace();
161     System.out.println("error: " + e);
162 }
163
164 public static void main(String[] args) {
165     new Pin().setVisible(true);
166 }

```

## conn.java

```

1 package bankmanagementsystem;
2
3 import java.sql.*;
4 import java.sql.DriverManager;
5 import java.sql.Statement;
6
7
8 public class conn{
9     Connection c;
10    Statement s;
11    public conn(){
12        try{
13            Class.forName("com.mysql.cj.jdbc.Driver");
14            c =DriverManager.getConnection("jdbc:mysql://localhost
15 :3306/atmbank","root","Musku@123");
16            s =c.createStatement();

```



```

17
18
19     }catch(Exception e){
20         System.out.println(e);
21     }
22 }
23 }

```

## signup.java

```

1 package bankmanagementsystem;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6 import java.util.*;
7
8 public class Signup extends JFrame implements ActionListener {
9
10     JLabel l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11, l12, l13, l14,
11         l15;
12     JTextField t1, t2, t3, t4, t5, t6, t7;
13     JRadioButton r1, r2, r3, r4, r5;
14     JButton b;
15     JComboBox c1, c2, c3;
16
17     Random ran = new Random();
18     long first4 = (ran.nextLong() % 9000L) + 1000L;
19     long first = Math.abs(first4);
20
21     Signup() {
22
23         setFont(new Font("System", Font.BOLD, 22));
24         Font f = getFont();
25         FontMetrics fm = getFontMetrics(f);
26         int x = fm.stringWidth("NEW ACCOUNT APPLICATION FORM");
27         int y = fm.stringWidth(" ");
28         int z = getWidth() - x;
29         int w = z / y;
30         String pad = "";
31         pad = String.format("%" + w + "s", pad);
32         setTitle(pad + "NEW ACCOUNT APPLICATION FORM");
33
34         l1 = new JLabel("APPLICATION FORM NO. " + first);
35         l1.setFont(new Font("Raleway", Font.BOLD, 38));
36
37         l2 = new JLabel("Page 1: Personal Details");
38         l2.setFont(new Font("Raleway", Font.BOLD, 22));
39
40         l3 = new JLabel("Name:");
41         l3.setFont(new Font("Raleway", Font.BOLD, 20));
42
43         l4 = new JLabel("Father's Name:");
44         l4.setFont(new Font("Raleway", Font.BOLD, 20));
45
46         l5 = new JLabel("Date of Birth:");
47         l5.setFont(new Font("Raleway", Font.BOLD, 20));

```



```

48     l6 = new JLabel("Gender:");
49     l6.setFont(new Font("Raleway", Font.BOLD, 20));
50
51     l7 = new JLabel("Email Address:");
52     l7.setFont(new Font("Raleway", Font.BOLD, 20));
53
54     l8 = new JLabel("Marital Status:");
55     l8.setFont(new Font("Raleway", Font.BOLD, 20));
56
57     l9 = new JLabel("Address:");
58     l9.setFont(new Font("Raleway", Font.BOLD, 20));
59
60     l10 = new JLabel("City:");
61     l10.setFont(new Font("Raleway", Font.BOLD, 20));
62
63     l11 = new JLabel("Pin Code:");
64     l11.setFont(new Font("Raleway", Font.BOLD, 20));
65
66     l12 = new JLabel("State:");
67     l12.setFont(new Font("Raleway", Font.BOLD, 20));
68
69     l13 = new JLabel("Date");
70     l13.setFont(new Font("Raleway", Font.BOLD, 14));
71
72     l14 = new JLabel("Month");
73     l14.setFont(new Font("Raleway", Font.BOLD, 14));
74
75     l15 = new JLabel("Year");
76     l15.setFont(new Font("Raleway", Font.BOLD, 14));
77
78     t1 = new JTextField();
79     t1.setFont(new Font("Raleway", Font.BOLD, 14));
80
81     t2 = new JTextField();
82     t2.setFont(new Font("Raleway", Font.BOLD, 14));
83
84     t3 = new JTextField();
85     t3.setFont(new Font("Raleway", Font.BOLD, 14));
86
87     t4 = new JTextField();
88     t4.setFont(new Font("Raleway", Font.BOLD, 14));
89
90     t5 = new JTextField();
91     t5.setFont(new Font("Raleway", Font.BOLD, 14));
92
93     t6 = new JTextField();
94     t6.setFont(new Font("Raleway", Font.BOLD, 14));
95
96     t7 = new JTextField();
97     t7.setFont(new Font("Raleway", Font.BOLD, 14));
98
99     b = new JButton("Next");
100    b.setFont(new Font("Raleway", Font.BOLD, 14));
101    b.setBackground(Color.BLACK);
102    b.setForeground(Color.WHITE);
103
104    r1 = new JRadioButton("Male");
105    r1.setFont(new Font("Raleway", Font.BOLD, 14));

```



```

106     r1.setBackground(Color.WHITE);
107
108     r2 = new JRadioButton("Female");
109     r2.setFont(new Font("Raleway", Font.BOLD, 14));
110     r2.setBackground(Color.WHITE);
111
112     r3 = new JRadioButton("Married");
113     r3.setFont(new Font("Raleway", Font.BOLD, 14));
114     r3.setBackground(Color.WHITE);
115
116     r4 = new JRadioButton("Unmarried");
117     r4.setFont(new Font("Raleway", Font.BOLD, 14));
118     r4.setBackground(Color.WHITE);
119
120     // Add "Other" radio button under gender
121     JRadioButton r5 = new JRadioButton("Other");
122     r5.setFont(new Font("Raleway", Font.BOLD, 14));
123     r5.setBackground(Color.WHITE);
124
125     // Gender radio buttons group
126     ButtonGroup genderGroup = new ButtonGroup();
127     genderGroup.add(r1);
128     genderGroup.add(r2);
129     genderGroup.add(r5); // Add the 'Other' button to the gender
group
130
131     // Marital radio buttons group (without "Other" option)
132     ButtonGroup maritalGroup = new ButtonGroup();
133     maritalGroup.add(r3);
134     maritalGroup.add(r4);
135
136     String date[] = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "
10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21
", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31"};
137     c1 = new JComboBox(date);
138     c1.setBackground(Color.WHITE);
139
140     String month[] = {"January", "February", "March", "April", "May
", "June", "July", "August", "September", "October", "November", "
December"};
141     c2 = new JComboBox(month);
142     c2.setBackground(Color.WHITE);
143
144     String year[] = {"1990", "1991", "1992", "1993", "1994", "1995"
, "1996", "1997", "1998", "1999", "2000", "2001", "2002", "2003", "
2004", "2005", "2006", "2007"};
145     c3 = new JComboBox(year);
146     c3.setBackground(Color.WHITE);
147
148     setLayout(null);
149     l1.setBounds(140, 20, 600, 40);
150     add(l1);
151
152     l2.setBounds(290, 70, 600, 30);
153     add(l2);
154
155     l3.setBounds(100, 140, 100, 30);
156     add(l3);

```



```

157
158     t1.setBounds(300, 140, 400, 30);
159     add(t1);
160
161     l4.setBounds(100, 190, 200, 30);
162     add(l4);
163
164     t2.setBounds(300, 190, 400, 30);
165     add(t2);
166
167     l5.setBounds(100, 240, 200, 30);
168     add(l5);
169
170     l13.setBounds(300, 240, 40, 30);
171     add(l13);
172
173     c1.setBounds(340, 240, 60, 30);
174     add(c1);
175
176     l14.setBounds(410, 240, 50, 30);
177     add(l14);
178
179     c2.setBounds(460, 240, 100, 30);
180     add(c2);
181
182     l15.setBounds(570, 240, 40, 30);
183     add(l15);
184
185     c3.setBounds(610, 240, 90, 30);
186     add(c3);
187
188     l6.setBounds(100, 290, 200, 30);
189     add(l6);
190
191     r1.setBounds(300, 290, 60, 30);
192     add(r1);
193
194     r2.setBounds(450, 290, 90, 30);
195     add(r2);
196
197     r5.setBounds(600, 290, 90, 30); // Position for the 'Other'
gender button
198     add(r5);
199
200     l7.setBounds(100, 340, 200, 30);
201     add(l7);
202
203     t3.setBounds(300, 340, 400, 30);
204     add(t3);
205
206     l8.setBounds(100, 390, 200, 30);
207     add(l8);
208
209     r3.setBounds(300, 390, 100, 30);
210     add(r3);
211
212     r4.setBounds(450, 390, 100, 30);
213     add(r4);

```



```

214
215     19.setBounds(100, 440, 200, 30);
216     add(19);
217
218     t4.setBounds(300, 440, 400, 30);
219     add(t4);
220
221     l10.setBounds(100, 490, 200, 30);
222     add(l10);
223
224     t5.setBounds(300, 490, 400, 30);
225     add(t5);
226
227     l11.setBounds(100, 540, 200, 30);
228     add(l11);
229
230     t6.setBounds(300, 540, 400, 30);
231     add(t6);
232
233     l12.setBounds(100, 590, 200, 30);
234     add(l12);
235
236     t7.setBounds(300, 590, 400, 30);
237     add(t7);
238
239     b.setBounds(620, 660, 80, 30);
240     add(b);
241
242     b.addActionListener(this);
243
244     getContentPane().setBackground(Color.WHITE);
245
246     setSize(850, 850);
247     setLocationRelativeTo(null);
248     setVisible(true);
249 }
250
251 public void actionPerformed(ActionEvent ae) {
252     String name = t1.getText();
253     String fname = t2.getText();
254
255     if (!name.matches("[a-zA-Z\\s]+$")) {
256         JOptionPane.showMessageDialog(null, "Name can only contain
letters and spaces.");
257         return;
258     }
259     if (!fname.matches("[a-zA-Z\\s]+$")) {
260         JOptionPane.showMessageDialog(null, "Father's Name can only
contain letters and spaces.");
261         return;
262     }
263
264     String date = (String) c1.getSelectedItem();
265     String month = (String) c2.getSelectedItem();
266     String year = (String) c3.getSelectedItem();
267
268     String gender = null;
269     if (r1.isSelected()) {

```



```

270         gender = "Male";
271     } else if (r2.isSelected()) {
272         gender = "Female";
273     } else if (r5.isSelected()) {
274         gender = "Other";
275     }
276
277     String email = t3.getText();
278     String marital = null;
279     if (r3.isSelected()) {
280         marital = "Married";
281     } else if (r4.isSelected()) {
282         marital = "Unmarried";
283     }
284
285     String address = t4.getText();
286     String city = t5.getText();
287     String pincode = t6.getText();
288     String state = t7.getText();
289
290     if (pincode.length() != 6 || !pincode.matches("\\d+")) {
291         JOptionPane.showMessageDialog(null, "Pin Code must be
292         exactly 6 digits.");
293         return;
294     }
295
296     try {
297         if (t6.getText().equals("")) {
298             JOptionPane.showMessageDialog(null, "Fill all the
299             required fields");
300         } else {
301             conn c1 = new conn();
302             String q1 = "insert into signup1 values('" + first + "
303             ',' + name + "','" + fname + "','" + date + "','" + month + "','" +
304             year + "','" + gender + "','" + email + "','" + marital + "','" +
305             address + "','" + city + "','" + pincode + "','" + state + "')";
306             c1.s.executeUpdate(q1);
307
308             setVisible(false);
309             new Signup2().setVisible(true);
310         }
311     } catch (Exception ex) {
312         ex.printStackTrace();
313     }
314 }
315
316 public static void main(String[] args) {
317     new Signup().setVisible(true);
318 }

```

## signup2.java

```

1
2 package bankmanagementsystem;
3
4 import java.awt.*;
5 import java.awt.event.*;

```



```

6 import javax.swing.*;
7 import java.sql.*;
8
9 public class Signup2 extends JFrame implements ActionListener {
10
11     JLabel l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11, l12;
12     JButton b;
13     JRadioButton r1, r2, r3, r4;
14     JTextField t1, t2, t3;
15     JComboBox c1, c2, c3, c4, c5;
16
17     ButtonGroup seniorGroup, existingGroup;
18
19     Signup2() {
20         setFont(new Font("System", Font.BOLD, 22));
21         Font f = getFont();
22         FontMetrics fm = getFontMetrics(f);
23         int x = fm.stringWidth("NEW ACCOUNT APPLICATION FORM - PAGE 2")
;
24         int y = fm.stringWidth(" ");
25         int z = getWidth() / 2 - (x / 2);
26         int w = z / y;
27         String pad = "";
28         pad = String.format("%" + w + "s", pad);
29         setTitle(pad + "NEW ACCOUNT APPLICATION FORM - PAGE 2");
30
31         l1 = new JLabel("Page 2: Additional Details");
32         l1.setFont(new Font("Raleway", Font.BOLD, 22));
33
34         l2 = new JLabel("Religion:");
35         l2.setFont(new Font("Raleway", Font.BOLD, 18));
36
37         l3 = new JLabel("Category:");
38         l3.setFont(new Font("Raleway", Font.BOLD, 18));
39
40         l4 = new JLabel("Income:");
41         l4.setFont(new Font("Raleway", Font.BOLD, 18));
42
43         l5 = new JLabel("Educational");
44         l5.setFont(new Font("Raleway", Font.BOLD, 18));
45
46         l11 = new JLabel("Qualification:");
47         l11.setFont(new Font("Raleway", Font.BOLD, 18));
48
49         l6 = new JLabel("Occupation:");
50         l6.setFont(new Font("Raleway", Font.BOLD, 18));
51
52         l7 = new JLabel("PAN Number:");
53         l7.setFont(new Font("Raleway", Font.BOLD, 18));
54
55         l8 = new JLabel("Aadhar Number:");
56         l8.setFont(new Font("Raleway", Font.BOLD, 18));
57
58         l9 = new JLabel("Senior Citizen:");
59         l9.setFont(new Font("Raleway", Font.BOLD, 18));
60
61         l10 = new JLabel("Existing Account:");
62         l10.setFont(new Font("Raleway", Font.BOLD, 18));

```



```

63     112 = new JLabel("Form No:");
64     112.setFont(new Font("Raleway", Font.BOLD, 13));
65
66     b = new JButton("Next");
67     b.setFont(new Font("Raleway", Font.BOLD, 14));
68     b.setBackground(Color.BLACK);
69     b.setForeground(Color.WHITE);
70
71     t1 = new JTextField();
72     t1.setFont(new Font("Raleway", Font.BOLD, 14));
73
74     t2 = new JTextField();
75     t2.setFont(new Font("Raleway", Font.BOLD, 14));
76
77     t3 = new JTextField();
78     t3.setFont(new Font("Raleway", Font.BOLD, 13));
79
80     r1 = new JRadioButton("Yes");
81     r1.setFont(new Font("Raleway", Font.BOLD, 14));
82     r1.setBackground(Color.WHITE);
83
84     r2 = new JRadioButton("No");
85     r2.setFont(new Font("Raleway", Font.BOLD, 14));
86     r2.setBackground(Color.WHITE);
87
88     r3 = new JRadioButton("Yes");
89     r3.setFont(new Font("Raleway", Font.BOLD, 14));
90     r3.setBackground(Color.WHITE);
91
92     r4 = new JRadioButton("No");
93     r4.setFont(new Font("Raleway", Font.BOLD, 14));
94     r4.setBackground(Color.WHITE);
95
96
97     seniorGroup = new ButtonGroup();
98     seniorGroup.add(r1);
99     seniorGroup.add(r2);
100
101     existingGroup = new ButtonGroup();
102     existingGroup.add(r3);
103     existingGroup.add(r4);
104
105     String religion[] = {"Hindu", "Muslim", "Sikh", "Christian", "
106 Other"};
107     c1 = new JComboBox(religion);
108     c1.setBackground(Color.WHITE);
109     c1.setFont(new Font("Raleway", Font.BOLD, 14));
110
111     String category[] = {"General", "OBC", "SC", "ST", "Other"};
112     c2 = new JComboBox(category);
113     c2.setBackground(Color.WHITE);
114     c2.setFont(new Font("Raleway", Font.BOLD, 14));
115
116     String income[] = {"Null", "<1,50,000", "<2,50,000", "<5,00,000
117 ", "Upto 10,00,000", "Above 10,00,000"};
118     c3 = new JComboBox(income);
119     c3.setBackground(Color.WHITE);

```



```

119         c3.setFont(new Font("Raleway", Font.BOLD, 14));
120
121         String education[] = {"Non-Graduate", "Graduate", "Post-
Graduate", "Doctorate", "Others"};
122         c4 = new JComboBox(education);
123         c4.setBackground(Color.WHITE);
124         c4.setFont(new Font("Raleway", Font.BOLD, 14));
125
126         String occupation[] = {"Salaried", "Self-Employed", "Business",
"Student", "Retired", "Others"};
127         c5 = new JComboBox(occupation);
128         c5.setBackground(Color.WHITE);
129         c5.setFont(new Font("Raleway", Font.BOLD, 14));
130
131         setLayout(null);
132
133         l12.setBounds(700, 10, 60, 30);
134         add(l12);
135
136         t3.setBounds(760, 10, 60, 30);
137         add(t3);
138
139         l11.setBounds(280, 30, 600, 40);
140         add(l11);
141
142         l12.setBounds(100, 120, 100, 30);
143         add(l12);
144
145         c1.setBounds(350, 120, 320, 30);
146         add(c1);
147
148         l13.setBounds(100, 170, 100, 30);
149         add(l13);
150
151         c2.setBounds(350, 170, 320, 30);
152         add(c2);
153
154         l14.setBounds(100, 220, 100, 30);
155         add(l14);
156
157         c3.setBounds(350, 220, 320, 30);
158         add(c3);
159
160         l15.setBounds(100, 270, 150, 30);
161         add(l15);
162
163         c4.setBounds(350, 270, 320, 30);
164         add(c4);
165
166         l111.setBounds(100, 290, 150, 30);
167         add(l111);
168
169         l16.setBounds(100, 340, 150, 30);
170         add(l16);
171
172         c5.setBounds(350, 340, 320, 30);
173         add(c5);
174

```



```

175     l7.setBounds(100, 390, 150, 30);
176     add(l7);
177
178     t1.setBounds(350, 390, 320, 30);
179     add(t1);
180
181     l8.setBounds(100, 440, 180, 30);
182     add(l8);
183
184     t2.setBounds(350, 440, 320, 30);
185     add(t2);
186
187     l9.setBounds(100, 490, 150, 30);
188     add(l9);
189
190     r1.setBounds(350, 490, 100, 30);
191     add(r1);
192
193     r2.setBounds(460, 490, 100, 30);
194     add(r2);
195
196     l10.setBounds(100, 540, 180, 30);
197     add(l10);
198
199     r3.setBounds(350, 540, 100, 30);
200     add(r3);
201
202     r4.setBounds(460, 540, 100, 30);
203     add(r4);
204
205     b.setBounds(570, 600, 100, 30);
206     add(b);
207
208     b.addActionListener(this);
209
210     getContentPane().setBackground(Color.gray);
211     setSize(850, 850);
212     setLocationRelativeTo(null);
213     setVisible(true);
214 }
215
216 public void actionPerformed(ActionEvent ae) {
217     String religion = (String) c1.getSelectedItem();
218     String category = (String) c2.getSelectedItem();
219     String income = (String) c3.getSelectedItem();
220     String education = (String) c4.getSelectedItem();
221     String occupation = (String) c5.getSelectedItem();
222
223     String pan = t1.getText();
224     String aadhar = t2.getText();
225
226     String senior = "";
227     if (r1.isSelected()) {
228         senior = "Yes";
229     } else if (r2.isSelected()) {
230         senior = "No";
231     }
232

```



```

233     String existing = "";
234     if (r3.isSelected()) {
235         existing = "Yes";
236     } else if (r4.isSelected()) {
237         existing = "No";
238     }
239
240     String formno = t3.getText();
241
242
243     if (!pan.matches("\\d{10}")) {
244         JOptionPane.showMessageDialog(null, "PAN Number must be
245         exactly 10 digits and contain only numbers.");
246         return;
247     }
248
249     if (!aadhar.matches("\\d{12}")) {
250         JOptionPane.showMessageDialog(null, "Aadhar Number must be
251         exactly 12 digits and contain only numbers.");
252         return;
253     }
254
255     try {
256         if (t2.getText().equals("")) {
257             JOptionPane.showMessageDialog(null, "Fill all the
258             required fields");
259         } else {
260             conn c1 = new conn();
261             String q1 = "insert into signup2 values('" + formno + "
262             ',' + religion + "',' + category + "',' + income + "',' +
263             education + "',' + occupation + "',' + pan + "',' + aadhar + "',' +
264             senior + "',' + existing + '');"
265             c1.s.executeUpdate(q1);
266
267             new Signup3().setVisible(true);
268             setVisible(false);
269         }
270     } catch (Exception ex) {
271         ex.printStackTrace();
272     }
273 }

```

## signup3.java

```

1
2 package bankmanagementsystem;
3
4 import java.awt.*;
5 import java.awt.event.*;
6 import javax.swing.*;
7 import java.sql.*;
8 import java.util.*;

```



```

9
10 public class Signup3 extends JFrame implements ActionListener {
11
12     JLabel l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11;
13     JRadioButton r1, r2, r3, r4;
14     JButton b1, b2;
15     JCheckBox c1, c2, c3, c4, c5, c6, c7;
16     JTextField t1;
17
18     Signup3() {
19
20         setFont(new Font("System", Font.BOLD, 22));
21         Font f = getFont();
22         FontMetrics fm = getFontMetrics(f);
23         int x = fm.stringWidth("NEW ACCOUNT APPLICATION FORM - PAGE 3");
24     ;
25         int y = fm.stringWidth(" ");
26         int z = getWidth() / 2 - (x / 2);
27         int w = z / y;
28         String pad = "";
29         pad = String.format("%" + w + "s", pad);
30         setTitle(pad + "NEW ACCOUNT APPLICATION FORM - PAGE 3");
31
32         l1 = new JLabel("Page 3: Account Details");
33         l1.setFont(new Font("Raleway", Font.BOLD, 22));
34
35         l2 = new JLabel("Account Type:");
36         l2.setFont(new Font("Raleway", Font.BOLD, 18));
37
38         l3 = new JLabel("Account Number:");
39         l3.setFont(new Font("Raleway", Font.BOLD, 18));
40
41         l4 = new JLabel("XXXX-XXXX-XXXX-4184");
42         l4.setFont(new Font("Raleway", Font.BOLD, 18));
43
44         l5 = new JLabel("(Your 16-digit Account number)");
45         l5.setFont(new Font("Raleway", Font.BOLD, 12));
46
47         l6 = new JLabel("It would appear on ATM Card/Cheque Book and
48         Statements");
49         l6.setFont(new Font("Raleway", Font.BOLD, 12));
50
51         l7 = new JLabel("PIN:");
52         l7.setFont(new Font("Raleway", Font.BOLD, 18));
53
54         l8 = new JLabel("XXXX");
55         l8.setFont(new Font("Raleway", Font.BOLD, 18));
56
57         l9 = new JLabel("(4-digit password)");
58         l9.setFont(new Font("Raleway", Font.BOLD, 12));
59
60         l10 = new JLabel("Services Required:");
61         l10.setFont(new Font("Raleway", Font.BOLD, 18));
62
63         l11 = new JLabel("Form No:");
64         l11.setFont(new Font("Raleway", Font.BOLD, 14));
65
66         b1 = new JButton("Submit");

```



```

65     b1.setFont(new Font("Raleway", Font.BOLD, 14));
66     b1.setBackground(Color.BLACK);
67     b1.setForeground(Color.WHITE);
68
69     b2 = new JButton("Cancel");
70     b2.setFont(new Font("Raleway", Font.BOLD, 14));
71     b2.setBackground(Color.BLACK);
72     b2.setForeground(Color.WHITE);
73
74     c1 = new JCheckBox("ATM CARD");
75     c1.setBackground(Color.WHITE);
76     c1.setFont(new Font("Raleway", Font.BOLD, 16));
77
78     c2 = new JCheckBox("Internet Banking");
79     c2.setBackground(Color.WHITE);
80     c2.setFont(new Font("Raleway", Font.BOLD, 16));
81
82     c3 = new JCheckBox("Mobile Banking");
83     c3.setBackground(Color.WHITE);
84     c3.setFont(new Font("Raleway", Font.BOLD, 16));
85
86     c4 = new JCheckBox("EMAIL Alerts");
87     c4.setBackground(Color.WHITE);
88     c4.setFont(new Font("Raleway", Font.BOLD, 16));
89
90     c5 = new JCheckBox("Cheque Book");
91     c5.setBackground(Color.WHITE);
92     c5.setFont(new Font("Raleway", Font.BOLD, 16));
93
94     c6 = new JCheckBox("E-Statement");
95     c6.setBackground(Color.WHITE);
96     c6.setFont(new Font("Raleway", Font.BOLD, 16));
97
98     c7 = new JCheckBox("I hereby declare that the above entered
99 details are correct to the best of my knowledge.", true);
100    c7.setBackground(Color.WHITE);
101    c7.setFont(new Font("Raleway", Font.BOLD, 12));
102
103    r1 = new JRadioButton("Saving Account");
104    r1.setFont(new Font("Raleway", Font.BOLD, 16));
105    r1.setBackground(Color.WHITE);
106
107    r2 = new JRadioButton("Fixed Deposit Account");
108    r2.setFont(new Font("Raleway", Font.BOLD, 16));
109    r2.setBackground(Color.WHITE);
110
111    r3 = new JRadioButton("Current Account");
112    r3.setFont(new Font("Raleway", Font.BOLD, 16));
113    r3.setBackground(Color.WHITE);
114
115    r4 = new JRadioButton("Recurring Deposit Account");
116    r4.setFont(new Font("Raleway", Font.BOLD, 16));
117    r4.setBackground(Color.WHITE);
118
119    t1 = new JTextField();
120    t1.setFont(new Font("Raleway", Font.BOLD, 12));
121
122    setLayout(null);

```



```
122
123     l11.setBounds(700, 10, 70, 30);
124     add(l11);
125
126     t1.setBounds(770, 10, 40, 30);
127     add(t1);
128
129     l1.setBounds(280, 50, 400, 40);
130     add(l1);
131
132     l2.setBounds(100, 140, 200, 30);
133     add(l2);
134
135     r1.setBounds(100, 180, 150, 30);
136     add(r1);
137
138     r2.setBounds(350, 180, 300, 30);
139     add(r2);
140
141     r3.setBounds(100, 220, 250, 30);
142     add(r3);
143
144     r4.setBounds(350, 220, 250, 30);
145     add(r4);
146
147     l3.setBounds(100, 300, 200, 30);
148     add(l3);
149
150     l4.setBounds(330, 300, 250, 30);
151     add(l4);
152
153     l5.setBounds(100, 330, 200, 20);
154     add(l5);
155
156     l6.setBounds(330, 330, 500, 20);
157     add(l6);
158
159     l7.setBounds(100, 370, 200, 30);
160     add(l7);
161
162     l8.setBounds(330, 370, 200, 30);
163     add(l8);
164
165     l9.setBounds(100, 400, 200, 20);
166     add(l9);
167
168     l10.setBounds(100, 450, 200, 30);
169     add(l10);
170
171     c1.setBounds(100, 500, 200, 30);
172     add(c1);
173
174     c2.setBounds(350, 500, 200, 30);
175     add(c2);
176
177     c3.setBounds(100, 550, 200, 30);
178     add(c3);
179
```



```

180     c4.setBounds(350, 550, 200, 30);
181     add(c4);
182
183     c5.setBounds(100, 600, 200, 30);
184     add(c5);
185
186     c6.setBounds(350, 600, 200, 30);
187     add(c6);
188
189     c7.setBounds(100, 680, 600, 20);
190     add(c7);
191
192     b1.setBounds(300, 720, 100, 30);
193     add(b1);
194
195     b2.setBounds(420, 720, 100, 30);
196     add(b2);
197
198     getContentPane().setBackground(Color.WHITE);
199
200     setSize(850, 850);
201     setLocationRelativeTo(null); // Center the frame
202     setVisible(true);
203
204     b1.addActionListener(this);
205     b2.addActionListener(this);
206
207 }
208
209 public void actionPerformed(ActionEvent ae) {
210
211     String a = null;
212     if (r1.isSelected()) {
213         a = "Saving Account";
214     } else if (r2.isSelected()) {
215         a = "Fixed Deposit Account";
216     } else if (r3.isSelected()) {
217         a = "Current Account";
218     } else if (r4.isSelected()) {
219         a = "Recurring Deposit Account";
220     }
221
222     Random ran = new Random();
223     long first7 = (ran.nextLong() % 900000000L) + 504093600000000000L;
224     long first8 = Math.abs(first7);
225
226     long first3 = (ran.nextLong() % 9000L) + 1000L;
227     long first4 = Math.abs(first3);
228
229     String b = "";
230     if (c1.isSelected()) {
231         b = b + " ATM Card";
232     }
233     if (c2.isSelected()) {
234         b = b + " Internet Banking";
235     }
236     if (c3.isSelected()) {
237         b = b + " Mobile Banking";

```



```

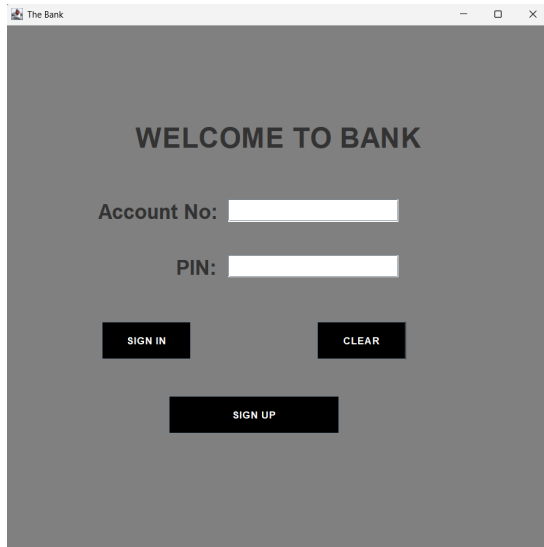
238     }
239     if (c4.isSelected()) {
240         b = b + " EMAIL Alerts";
241     }
242     if (c5.isSelected()) {
243         b = b + " Cheque Book";
244     }
245     if (c6.isSelected()) {
246         b = b + " E-Statement";
247     }
248
249     String c = t1.getText();
250
251     try {
252
253         if (ae.getSource() == b1) {
254
255             if (b.equals("")) {
256
257                 JOptionPane.showMessageDialog(null, "Fill all the
required fields");
258
259             } else {
260
261                 conn c1 = new conn();
262                 String q1 = "insert into signup3 values('" + c + "
','" + a + "','" + first8 + "','" + first4 + "','" + b + "')";
263                 String q2 = "insert into login values('" + first8 +
"','" + first4 + "')";
264                 String q3 = "insert into bank values('" + first4 +
"','0,0,0,'" + first8 + "','1)";
265
266                 c1.s.executeUpdate(q1);
267                 c1.s.executeUpdate(q2);
268                 c1.s.executeUpdate(q3);
269                 JOptionPane.showMessageDialog(null, "Account Number
: " + first8 + "\n Pin:" + first4);
270
271                 new Deposit().setVisible(true);
272                 setVisible(false);
273             }
274
275         } else if (ae.getSource() == b2) {
276             System.exit(0);
277         }
278
279     } catch (Exception ex) {
280         ex.printStackTrace();
281     }
282
283 }
284
285 public static void main(String[] args) {
286     new Signup3().setVisible(true);
287 }
288
289 }

```



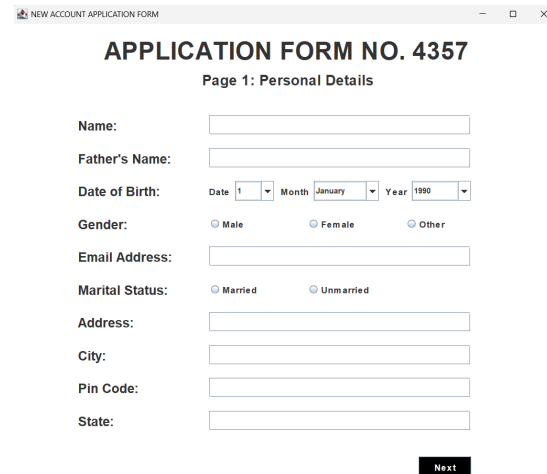
## 4. Output

Below are the output screenshots of the Bank Management System project showing different functionalities:



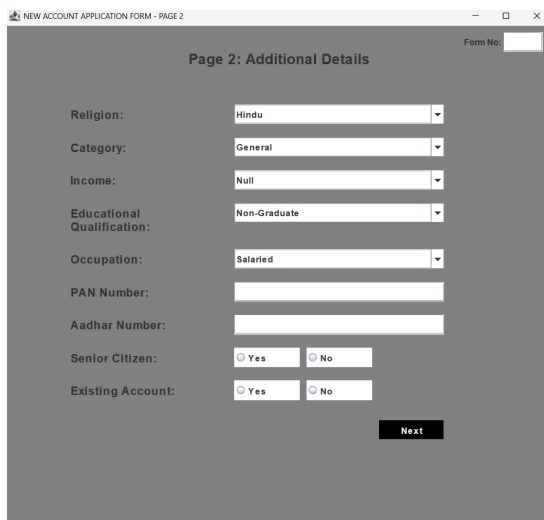
The screenshot shows a web browser window titled "The Bank". The page has a dark gray background with the text "WELCOME TO BANK" in white. Below this, there are two input fields: "Account No:" and "PIN:". To the right of each field is a white input box. Below the input fields, there are three buttons: "SIGN IN" and "CLEAR" are side-by-side, and "SIGN UP" is centered below them. All buttons are black with white text.

Figure 1: Login Page



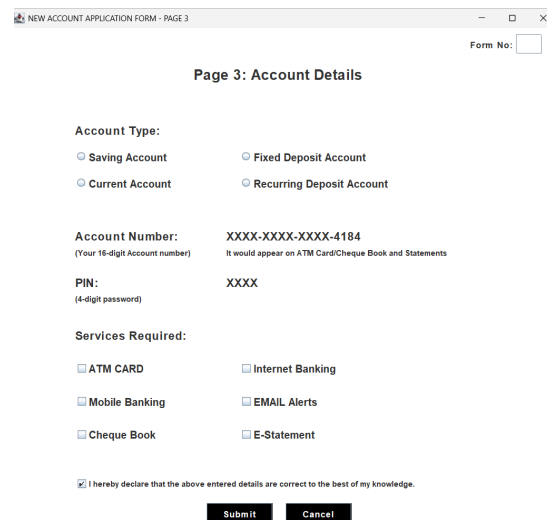
The screenshot shows a web browser window titled "NEW ACCOUNT APPLICATION FORM". The page has a white background with the text "APPLICATION FORM NO. 4357" in bold. Below this, it says "Page 1: Personal Details". The form includes the following fields: "Name:" (text input), "Father's Name:" (text input), "Date of Birth:" (Date, Month, Year dropdowns), "Gender:" (radio buttons for Male, Female, Other), "Email Address:" (text input), "Marital Status:" (radio buttons for Married, Unmarried), "Address:" (text input), "City:" (text input), "Pin Code:" (text input), and "State:" (text input). A "Next" button is at the bottom right.

Figure 2: Signup Page - Step 1



The screenshot shows a web browser window titled "NEW ACCOUNT APPLICATION FORM - PAGE 2". The page has a dark gray background with the text "Page 2: Additional Details" in white. The form includes the following fields: "Religion:" (dropdown menu), "Category:" (dropdown menu), "Income:" (dropdown menu), "Educational Qualification:" (dropdown menu), "Occupation:" (dropdown menu), "PAN Number:" (text input), "Aadhar Number:" (text input), "Senior Citizen:" (radio buttons for Yes, No), and "Existing Account:" (radio buttons for Yes, No). A "Next" button is at the bottom right.

Figure 3: Signup Page - Step 2



The screenshot shows a web browser window titled "NEW ACCOUNT APPLICATION FORM - PAGE 3". The page has a white background with the text "Page 3: Account Details". The form includes the following fields: "Account Type:" (radio buttons for Saving Account, Fixed Deposit Account, Current Account, Recurring Deposit Account), "Account Number:" (text input, with a note: "XXXX-XXXX-XXXX-4184 It would appear on ATM Card/Cheque Book and Statements"), "PIN:" (text input, with a note: "XXXX (4-digit password)"), "Services Required:" (checkboxes for ATM CARD, Internet Banking, Mobile Banking, EMAIL Alerts, Cheque Book, E-Statement), and a declaration checkbox: "I hereby declare that the above entered details are correct to the best of my knowledge." Below the declaration are "Submit" and "Cancel" buttons.

Figure 4: Signup Page - Step 3







## 5. MySQL Database

Below is a screenshot showing the MySQL database structure used in the Bank Management System project. It includes the necessary tables and columns for storing user credentials, account information, transactions, and more.

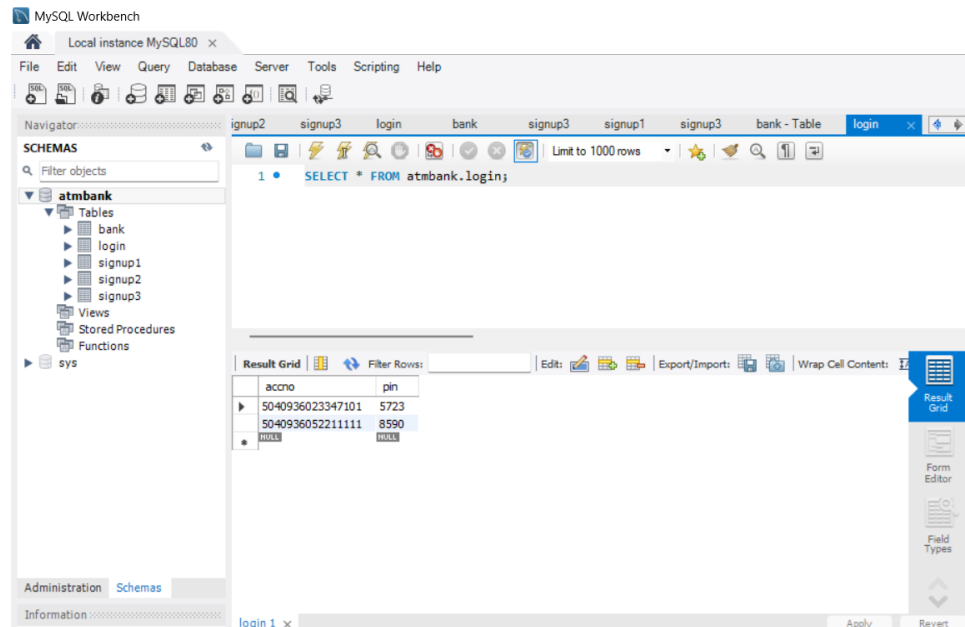


Figure 10: MySQL Database