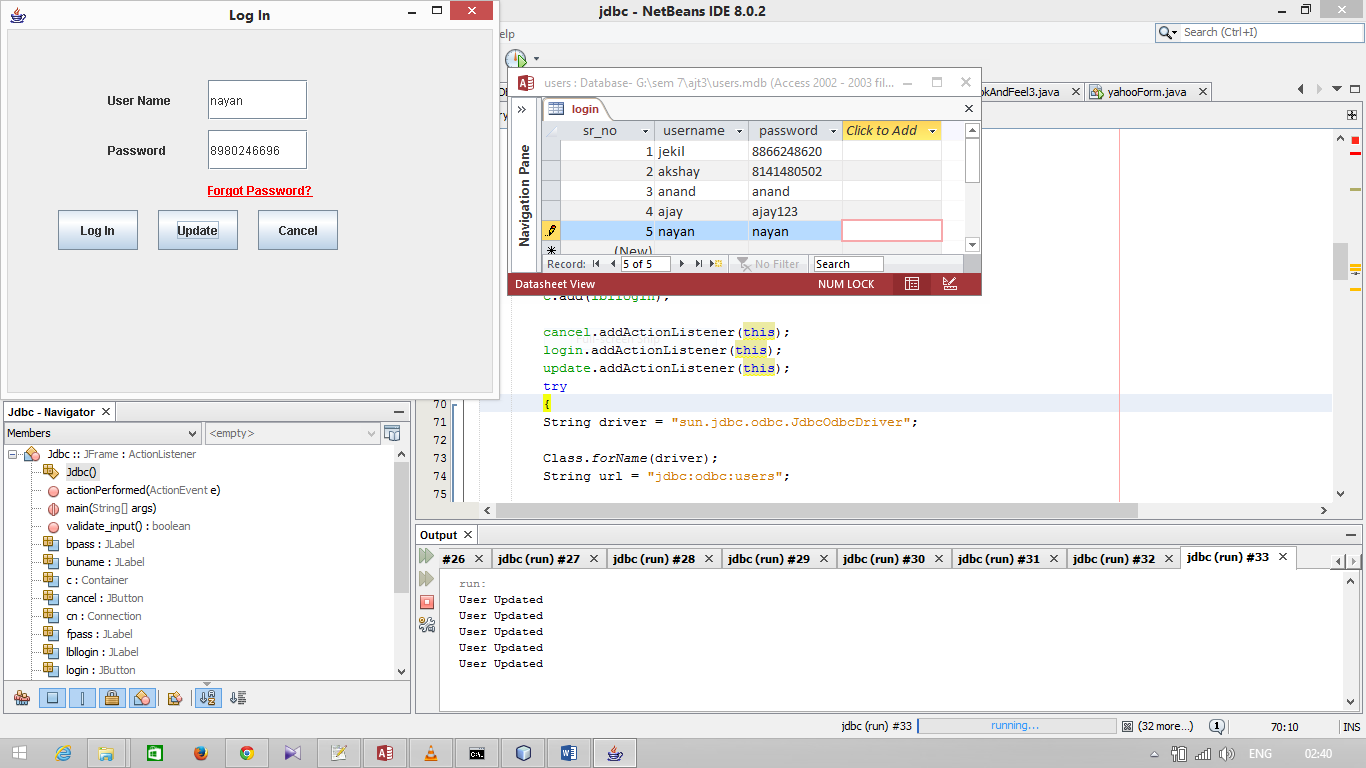
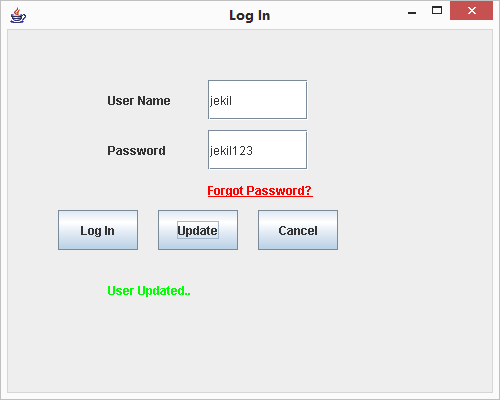
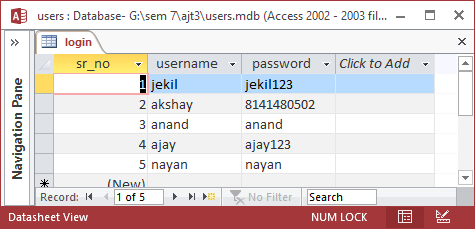
**Prepare a java application using jdbc-odbc bridge driver update login table with 5 records.**

1. import java.awt.\*;
2. import java.awt.event.\*;
3. import javax.swing.\*;
4. import java.sql.Connection;
5. import java.sql.DriverManager;
6. import java.sql.ResultSet;
7. import java.sql.Statement;
8. import java.sql.PreparedStatement;
9. public class Jdbc extends JFrame implements ActionListener
10. {
11. Connection cn;
12. Statement st;
13. ResultSet rs;
14. JTextField username, password;
15. JLabel uname,pass,buname,bpass,lbllogin,fpass;
16. JButton login,cancel,update;
17. Container c;
18. Jdbc()
19. {
20. c = this.getContentPane();
21. c.setLayout(null);
22. lbllogin= new JLabel("");
23. lbllogin.setBounds(100, 240, 200, 40);
24. username= new JTextField();
25. password=new JTextField();
26. fpass = new JLabel("<html><u>Forgot Password?</u></html>");
27. fpass.setForeground(Color.red);
28. fpass.setBounds(200,140,200,40);
29. uname = new JLabel("User Name ");
30. pass = new JLabel("Password");
31. buname = new JLabel("");
32. bpass = new JLabel("");
33. login = new JButton("Log In");
34. cancel = new JButton("Cancel");
35. update = new JButton("Update");
37. uname.setBounds(100,50,100,40);
38. username.setBounds(200, 50, 100, 40);
39. buname.setBounds(300,50,100,40);
40. pass.setBounds(100,100,100,40);
41. password.setBounds(200,100,100,40);
42. bpass.setBounds(300,100,100,40);
43. login.setBounds(50, 180, 80, 40);
44. cancel.setBounds(250, 180, 80, 40);
45. update.setBounds(150,180,80,40);
46. c.add(fpass);
47. c.add(update);
48. c.add(username);
49. c.add(uname);
50. c.add(password);
51. c.add(pass);
52. c.add(cancel);
53. c.add(login);
54. c.add(buname);
55. c.add(bpass);
56. c.add(lbllogin);
57. cancel.addActionListener(this);
58. login.addActionListener(this);
59. update.addActionListener(this);
60. try
61. {
62. String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
63. Class.forName(driver);
64. String url = "jdbc:odbc:users";
65. cn=DriverManager.getConnection(url);
66. st=cn.createStatement();
67. }
68. catch(Exception e)
69. {
70. System.out.println("Something going wrong with Database");
71. }
72. }
73. @Override
74. public void actionPerformed(ActionEvent e)
75. {
76. String btn=e.getActionCommand();
77. if(btn.equals("Log In"))
78. {
79. if(validate\_input())
80. {
81. try
82. {
83. String query1 = "select \* from login where username='" + username.getText() + "'";
84. String query = "select \* from login";
85. String u,p;
86. int i;
87. i=0;
88. rs = st.executeQuery(query);
89. while(rs.next())
90. {
91. u=rs.getString("username");
92. p=rs.getString("password");
93. if(u.equals(username.getText()) && p.equals(password.getText()))
94. {
95. System.out.println("Logged In ");
96. i=1;
97. }
98. }
99. if(i==1)
100. {
101. lbllogin.setText("Login Success..");
102. lbllogin.setForeground(Color.green);
103. }
104. else
105. {
106. lbllogin.setText("Please Enter Valied Credential..");
107. lbllogin.setForeground(Color.red);
108. }
109. }
110. catch(Exception lg)
111. {
112. System.out.println(lg.getMessage());
113. }
114. }
115. }
116. if(btn.equals("Cancel"))
117. {
118. username.setText("");
119. password.setText("");
120. buname.setText("");
121. bpass.setText("");
122. lbllogin.setText("");
123. }
124. if(btn.equals("Update"))
125. {
126. if(validate\_input())
127. {
128. try
129. {
130. String query = "update login SET password = ? where username = ?";
131. PreparedStatement pst = cn.prepareStatement(query);
132. pst.setString(2,username.getText());
133. pst.setString(1,password.getText());
134. int i = pst.executeUpdate();
135. System.out.println("User Updated");
136. lbllogin.setText("User Updated..");
137. lbllogin.setForeground(Color.green);
139. }
140. catch(Exception es)
141. {
142. System.out.println(es);
143. }
144. }
145. }
146. }
147. public boolean validate\_input()
148. {
149. String unamev, passv;
150. unamev = username.getText();
151. passv = password.getText();
152. if(unamev.equals(""))
153. {
154. buname.setText("Enter User Name");
155. buname.setForeground(Color.red);
156. return false;
157. }
158. if(passv.equals(""))
159. {
160. bpass.setText("Enter Password");
161. bpass.setForeground(Color.red);
162. return false;
163. }
164. return true;
165. }
166. public static void main(String[] args) {
167. Jdbc yf = new Jdbc();
168. yf.setSize(500,400);
169. yf.setTitle("Log In");
170. yf.setVisible(true);
171. }
173. }

**OUTPUT**

****

** **