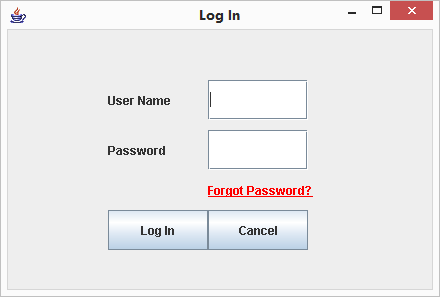
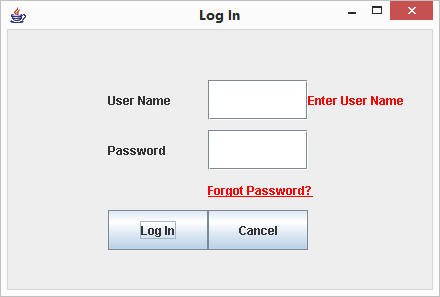
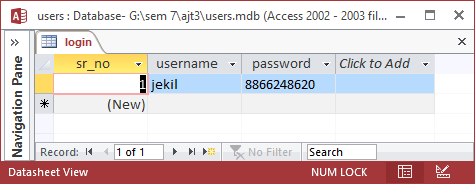
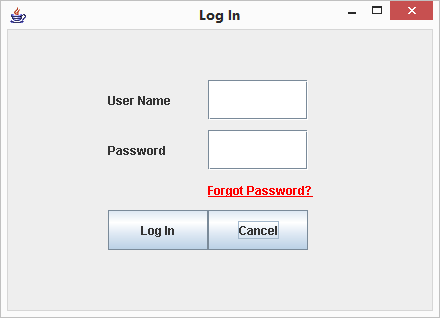
**Prepare a java application using jdbc-odbc bridge driver to create login form that includes username, password, login, cancel button and forgot password link.**

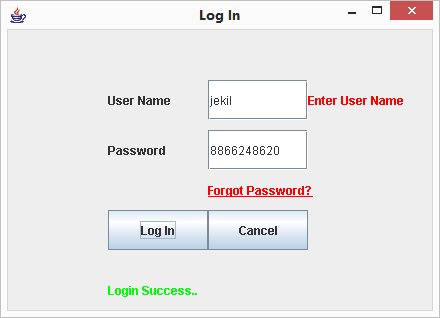
1. import java.awt.\*;
2. import java.awt.event.\*;
3. import javax.swing.\*;
4. import java.sql.Connection;
5. import java.sql.DriverManager;
6. import java.sql.ResultSet;
7. import java.sql.Statement;
8. public class JdbcLogin extends JFrame implements ActionListener
9. {
10. Connection cn;
11. Statement st;
12. ResultSet rs;
13. JTextField username, password;
14. JLabel uname,pass,buname,bpass,lbllogin,fpass;
15. JButton login,cancel;
16. Container c;
17. JdbcLogin()
18. {
19. c = this.getContentPane();
20. c.setLayout(null);
21. lbllogin= new JLabel("");
22. lbllogin.setBounds(100, 240, 200, 40);
23. username= new JTextField();
24. password=new JTextField();
25. fpass = new JLabel("<html><u>Forgot Password?</u></html>");
26. fpass.setForeground(Color.red);
27. fpass.setBounds(200,140,200,40);
28. uname = new JLabel("User Name ");
29. pass = new JLabel("Password");
30. buname = new JLabel("");
31. bpass = new JLabel("");
32. login = new JButton("Log In");
33. cancel = new JButton("Cancel");
34. uname.setBounds(100,50,100,40);
35. username.setBounds(200, 50, 100, 40);
36. buname.setBounds(300,50,100,40);
37. pass.setBounds(100,100,100,40);
38. password.setBounds(200,100,100,40);
39. bpass.setBounds(300,100,100,40);
40. login.setBounds(100, 180, 100, 40);
41. cancel.setBounds(200, 180, 100, 40);
42. c.add(fpass);
43. c.add(username);
44. c.add(uname);
45. c.add(password);
46. c.add(pass);
47. c.add(cancel);
48. c.add(login);
49. c.add(buname);
50. c.add(bpass);
51. c.add(lbllogin);
52. cancel.addActionListener(this);
53. login.addActionListener(this);
54. try
55. {
56. String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
57. Class.forName(driver);
58. String url = "jdbc:odbc:users";
59. cn=DriverManager.getConnection(url);
60. st=cn.createStatement();
61. }
62. catch(Exception e)
63. {
64. System.out.println("Something going wrong with Database");
65. }
66. }
67. @Override
68. public void actionPerformed(ActionEvent e)
69. {
70. String btn=e.getActionCommand();
71. if(btn.equals("Log In"))
72. {
73. if(validate\_input())
74. {
75. try
76. {
77. System.out.println("Hello");
78. String query1 = "select \* from login where username='" + username.getText() + "'";
79. String query = "select \* from login";
80. String u,p;
81. int i;
82. i=0;
83. rs = st.executeQuery(query);
84. while(rs.next())
85. {
86. u=rs.getString("username");
87. p=rs.getString("password");
88. if(u.equals(username.getText()) && p.equals(password.getText()))
89. {
90. System.out.println("Logged In ");
91. i=1;
92. }
93. }
94. if(i==1)
95. {
96. lbllogin.setText("Login Success..");
97. lbllogin.setForeground(Color.green);
98. }
99. else
100. {
101. lbllogin.setText("Please Enter Valied Credential..");
102. lbllogin.setForeground(Color.red);
103. }
104. }
105. catch(Exception lg)
106. {
107. System.out.println(lg.getMessage());
108. }
109. }
110. }
111. if(btn.equals("Cancel"))
112. {
113. username.setText("");
114. password.setText("");
115. buname.setText("");
116. bpass.setText("");
117. lbllogin.setText("");
118. }
119. }
120. public boolean validate\_input()
121. {
122. String unamev, passv;
123. unamev = username.getText();
124. passv = password.getText();
125. if(unamev.equals(""))
126. {
127. buname.setText("Enter User Name");
128. buname.setForeground(Color.red);
129. return false;
130. }
131. if(passv.equals(""))
132. {
133. bpass.setText("Enter Password");
134. bpass.setForeground(Color.red);
135. return false;
136. }
137. return true;
138. }
139. public static void main(String[] args) {
140. JdbcLogin yf = new JdbcLogin();
141. yf.setSize(500,400);
142. yf.setTitle("Log In");
143. yf.setVisible(true);
144. }
146. }

**OUTPUT**

** **

****

****

****