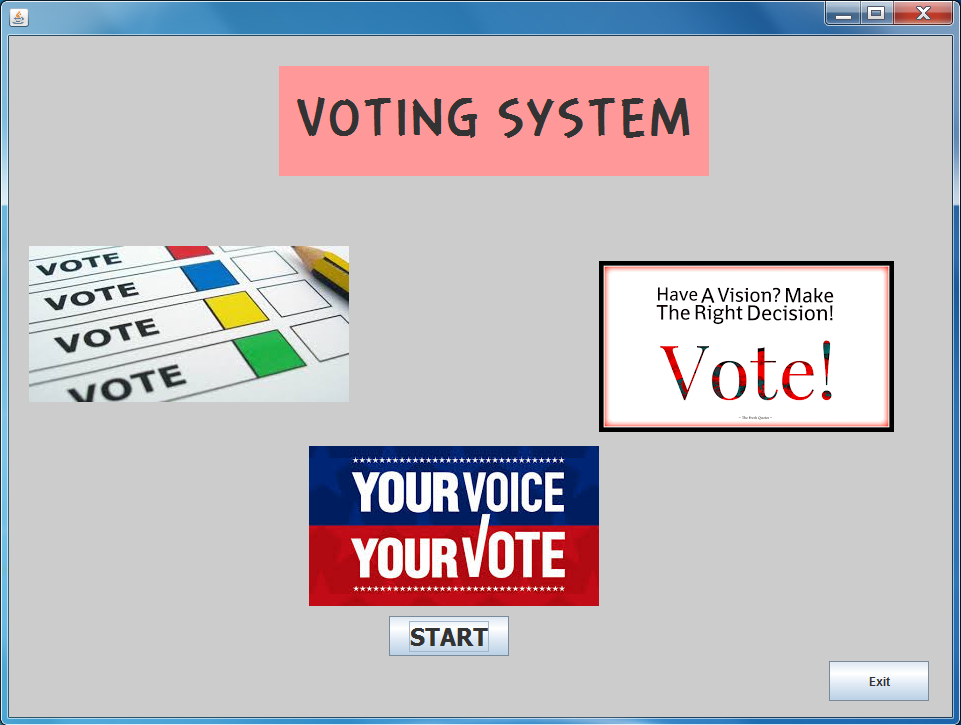
**INDEX**

|  |  |  |
| --- | --- | --- |
| ***S.No.*** | ***Title*** | ***Page No.*** |
|  | **Frame Forms in Netbeans IDE** |  |
| 1. | Start |  |
| 2. | User login form |  |
| 3. | Vote |  |
| 4. | Admin login form |  |
| 5. | Edit form |  |
| 6. | Form to add a candidate |  |
| 7. | Form to add a voter |  |
| 8. | Form to delete a candidate |  |
| 9. | Form to delete a voter |  |
| 10. | Form to update a candidate |  |
| 11. | Form to update a voter |  |
| 12. | Result |  |
|  | **Database in MySQL** |  |
| 13. | Creating database |  |
| 14. | Creating tables |  |
| 15. | Examples of data in tables |  |
| 16. | Desc commands of tables |  |

***Frame Forms in Netbeans***

1. **START**

****

**Code for Start button:-**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

new loginform().setVisible(true);

this.dispose();

}

**Code for Exit button:-**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);

}

1. **USER LOGIN FORM**

****

**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jLabel | 3 | jLabel1  jLabel2  jLabel3 | USER LOGIN  Username  Password |
| jTextField | 2 | tf1  tf2 | Nil  Nil |
| jButton | 3 | jButton1  jButton2  jButton3 | Admin Login  Log in  Exit |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

**Code for ActionPerformed event of jButton2 (Log in):-**

private void jButton2ActionPerformed(java.awt.event.ActionEventevt) {

try{

String uname= tf1.getText();

String pw= tf2.getText();

if((uname.length()!=0)&&(pw.length()!=0))

{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query1="select voted from loginfo where username='" + uname +"' and password ='" +pw+ "'";

ResultSetrs=stmt.executeQuery(query1);

rs.next();

if(rs.first()==true)

{

if(rs.getString("voted").equals("no"))

{

new vote().setVisible(true);

tf1.setText("");

tf2.setText("");

String query2= "update loginfo set flag='on' where username='" + uname +"' and password ='" +pw+ "'";

stmt.executeUpdate(query2);

}

else{

JOptionPane.showMessageDialog(null,"You have already voted. Access denied");

tf1.setText("");

tf2.setText("");

}

}

else{

JOptionPane.showMessageDialog(null,"Access denied");

tf1.setText("");

tf2.setText("");

}

rs.close();

stmt.close();

conn.close();}

else JOptionPane.showMessageDialog(null,"Empty field, please insert values.");

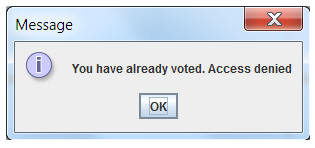
}

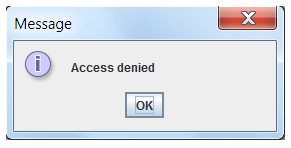
catch(Exception e){

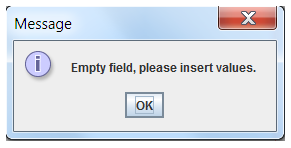
JOptionPane.showMessageDialog(null,e);

}

}







**Code for ActionPerformed event of jButton3 (Exit):-**

private void jButton3ActionPerformed(java.awt.event.ActionEventevt)

{

System.exit(0);

}

**Code for ActionPerformed event of jButton1(Admin login):-**

private void jButton1ActionPerformed(java.awt.event.ActionEventevt) {

new adminform().setVisible(true);

this.dispose();

}

**Code for FocusLost event of tf1:-**

private void tf1FocusLost(java.awt.event.FocusEventevt) {

String s=tf1.getText();

if(s.length()!=0)

{

boolean flag=false;

for(inti=0;i<=s.length()-1;i++)

{

if(((s.charAt(i)>=65&&s.charAt(i)<=90)||(s.charAt(i)>=97

&&s.charAt(i)<=122)||s.charAt(i)==32)==false)

{

flag=false;

break;}

else {flag=true;

}

}

if(flag==false)

{

JOptionPane.showMessageDialog(null,"You cannot enter characters

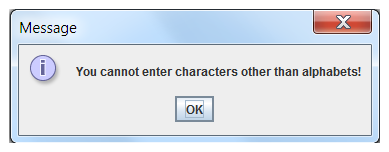
other than alphabets!");

tf1.setText("");

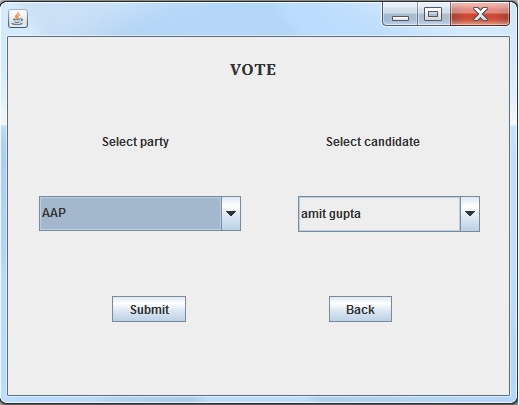
}

}

}



1. **VOTE**

****

**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jLabel | 3 | jLabel1  jLabel2  jLabel3 | VOTE  Select party  Select candidate |
| jComboBox | 2 | jComboBox1  jComboBox2 |  |
| jButton | 2 | jButton1  jButton2 | Submit  Back |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

**Code for WindowActivatedevent:-**

private void formWindowActivated(java.awt.event.WindowEventevt) {

try{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query="select distinct(party\_name) from vote";

ResultSetrs=stmt.executeQuery(query);

while(rs.next())

{

String pname = rs.getString("party\_name");

jComboBox1.addItem(pname);

}

}

catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

**Code for ActionPerformed event of jButton1(Submit):-**

private void jButton1ActionPerformed(java.awt.event.ActionEventevt) {

try{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

Object a=jComboBox2.getSelectedItem();

String query1="update vote set votes=votes+1 where candidate\_name='"+a+"'";

stmt.executeUpdate(query1);

String query2="update loginfo set voted ='yes' where flag='on'";

stmt.executeUpdate(query2);

JOptionPane.showMessageDialog(null,"You have voted successfully.Thank you!");

String query3="update loginfo set flag='off' where flag='on'";

stmt.executeUpdate(query3);

this.dispose();

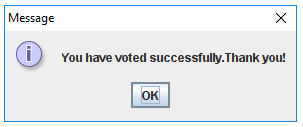
}

catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}



**Code for ActionPerformed event of jButton2(Back):**

private void jButton2ActionPerformed(java.awt.event.ActionEventevt) {

try{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query3="update loginfo set flag='off' where flag='on'";

stmt.executeUpdate(query3);

}

catch(Exception e){

JOptionPane.showMessageDialog(null,e);}

this.dispose();

new loginform().setVisible(true);

}

**Code for ActionPerformed event of jComboBox1(for party):-**

private void jComboBox1ActionPerformed(java.awt.event.ActionEventevt) {

try{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

Object pname = jComboBox1.getSelectedItem();

String query="select candidate\_name from vote where party\_name = '"+pname+"' ";

ResultSetrs=stmt.executeQuery(query);

jComboBox2.removeAllItems();

while(rs.next())

{

String c = rs.getString("candidate\_name");

jComboBox2.addItem(c);

}

}

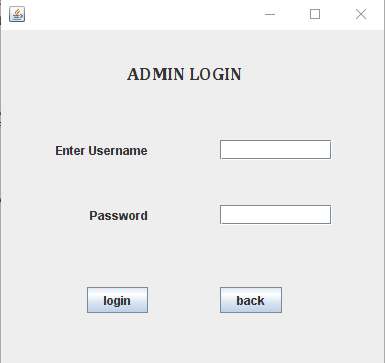
catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

1. ADMIN LOGIN FORM



**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jLabel | 3 | jLabel1  jLabel2  jLabel3 | ADMIN LOGIN  Enter Username  Password |
| jTextField | 2 | jTextField1  jTextField2 | Nil  Nil |
| jButton | 2 | jButton1  jButton2 | login  Back |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

**Code for ActionPerformed event of jButton2(Back):-**

private void jButton2ActionPerformed(java.awt.event.ActionEventevt) {

this.dispose();

new loginform().setVisible(true);

}

**Code for ActionPerformed event of jButton1(login):-**

private void jButton1ActionPerformed(java.awt.event.ActionEventevt) {

try{

String uname= jTextField1.getText();

String pw= jTextField2.getText();

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query="select \* from admin where username='" + uname +"' and password ='" +pw+ "'";

ResultSetrs=stmt.executeQuery(query);

rs.next();

if((uname.length()!=0)&&(pw.length()!=0))

{

if(rs.first()==true)

{

if((rs.getString("username").equals("i"))&&(rs.getString("password").equals("i")))

{

new admindetails().setVisible(true);

this.dispose();

}

else JOptionPane.showMessageDialog(null,"Access denied!!");

}

else JOptionPane.showMessageDialog(null,"Access denied!!");

rs.close();

stmt.close();

conn.close();

}

else JOptionPane.showMessageDialog(null,"Empty fields, please insert values.");

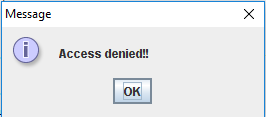
}

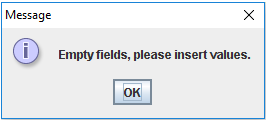
catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}



****

**Code for FocusLost event of jTextField1 :-**

private void jTextField1FocusLost(java.awt.event.FocusEventevt) {

String s=jTextField1.getText();

if(s.length()!=0)

{

boolean flag=false;

for(inti=0;i<=s.length()-1;i++){

if(((s.charAt(i)>=65&&s.charAt(i)<=90)||(s.charAt(i)>=97 &&s.charAt(i)<=122)||s.charAt(i)==32)==false)

{

flag=false;

break;

}

else flag=true;

}

if(flag==false)

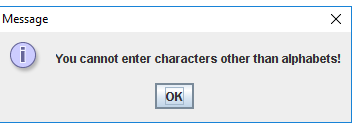
{JOptionPane.showMessageDialog(null,"You cannot enter characters other than alphabets!");

jTextField1.setText("");

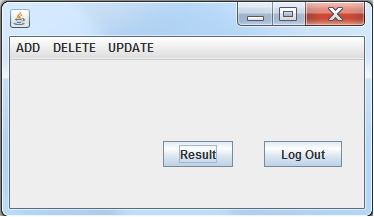
}

}

}



1. EDIT FORM



**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jMenuBar | 1 | jMenuBar1 | Nil |
| jMenu | 3 | jMenu1  jMenu2  jMenu3 | ADD  DELETE  UPDATE |
| jMenuItem | 6 | jMenuItem1  jMenuItem2  jMenuItem3  jMenuItem4  jMenuItem5  jMenuItem6 | Candidate  Voter  Candidate  Voter  Candidate  Voter |
| jButton | 2 | jButton1  jButton2 | Log out  Result |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

**Code for ActionPerformed event of jMenuItem1:-**

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt)

{

new addcandidate().setVisible(true);

}

**Code for ActionPerformed event of jMenuItem2:-**

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt){

new addvoter().setVisible(true);

}

**Code for ActionPerformed event of jMenuItem3:-**

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {

new deletecandidate().setVisible(true); }

**Code for ActionPerformed event of jMenuItem4:-**

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt)

{

new deletevoter().setVisible(true);

}

**Code for ActionPerformed event of jMenuItem5:-**

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt)

{

new updatecandidate().setVisible(true);

}

**Code for ActionPerformed event of jMenuItem6:-**

private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt)

{

new updatevoter().setVisible(true);

}

**Code for WindowActivated event :-**

private void formWindowActivated(java.awt.event.WindowEvent evt)

{

setExtendedState(6);

}

**Code for ActionPerformed event of jButton1(Log out):-**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

{

this.dispose();

new adminform().setVisible(true);

}

**Code for ActionPerformed event of jButton1(Result):-**

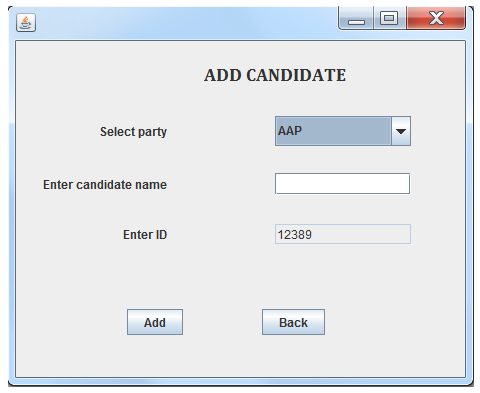
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)

{

new result().setVisible(true);

}

1. FORM TO ADD A CANDIDATE

****

**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jLabel | 4 | jLabel1  jLabel2  jLabel3  jLabel4 | ADD CANDIDATE  Select party  Enter candidate name  Enter ID |
| jTextField | 2 | jTextField2  jTextField3 | Nil  (ID at run time,uneditable) |
| jButton | 2 | jButton1  jButton2 | Add  Back |
| jComboBox | 1 | jComboBox1 | Nil |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

**Code for ActionPerformed event of jButton1:-**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

{

try{

Object pname = jComboBox1.getSelectedItem();

String candidate= jTextField2.getText();

String idd=jTextField3.getText();

int id = Integer.parseInt(idd);

if((candidate.length()!=0)&&(idd.length()!=0))

{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query="insert into vote values('"+candidate+"','"+pname+"',"+id+",0)";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(null,"Your changes have been saved successfully!");

new admindetails().setVisible(true);

this.dispose();

}

else JOptionPane.showMessageDialog(null,"Empty field, please insert values.");

}

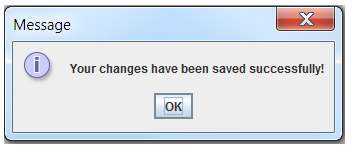
catch(Exception e)

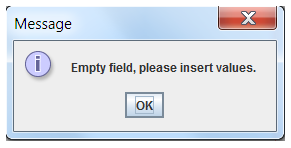
{

JOptionPane.showMessageDialog(null,e);

}

}



****

**Code for ActionPerformed event of jButton2:-**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)

{

this.dispose();

new admindetails().setVisible(true);

}

**Code for FocusLost event of jTextField2:-**

private void jTextField2FocusLost(java.awt.event.FocusEvent evt)

{

String s=jTextField2.getText();

if(s.length()!=0){

boolean flag=false;

for(int i=0;i<=s.length()-1;i++){

if(((s.charAt(i)>=65&&s.charAt(i)<=90)||(s.charAt(i)>=97 && s.charAt(i)<=122)||s.charAt(i)==32)==false){

flag=false;

break;}

else {

flag=true;

}

}

if(flag==false)

{

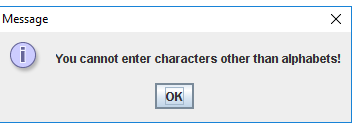
JOptionPane.showMessageDialog(null,"You cannot enter characters other than alphabets!");

jTextField2.setText("");

}

}

}



**Code for WindowActivated event :-**

private void formWindowActivated(java.awt.event.WindowEvent evt)

{

try

{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query="select distinct(party\_name) as 'p' from vote";

ResultSet rs=stmt.executeQuery(query);

while(rs.next())

{

String pname = rs.getString("p");

jComboBox1.addItem(pname);

}

}

catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

**Code for ActionPerformed event of jComboBox1:-**

private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt)

{

try{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query="select max(id) as 'v' from vote";

ResultSet rs=stmt.executeQuery(query);

rs.next();

int max = rs.getInt("v");

int q = max+1;

jTextField3.setText(""+q);

}

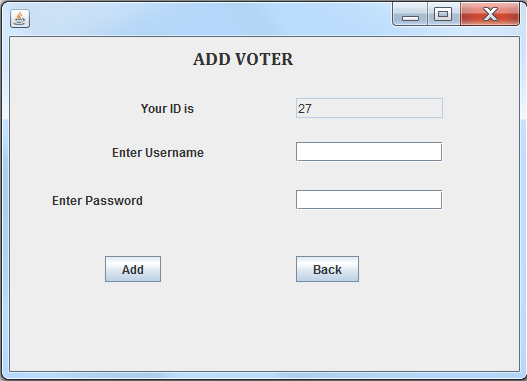
catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

1. FORM TO ADD A VOTER

****

**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jLabel | 4 | jLabel1  jLabel2  jLabel3  jLabel4 | ADD VOTER  Your ID is  Enter Username  Enter Password |
| jTextField | 3 | jTextField1  jTextField2  jTextField3 | (ID at run time,uneditable)  Nil  Nil |
| jButton | 2 | jButton1  jButton2 | Add  Back |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

**Code for ActionPerformed event of jButton1:-**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

{

try{

String uname= jTextField2.getText();

String pwd= jTextField3.getText();

int faltu= Integer.parseInt(jTextField1.getText());

if((uname.length()!=0)&&(pwd.length()!=0))

{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query1="select \* from loginfo where username='" + uname +"' and password ='" +pwd+ "'";

ResultSet rs=stmt.executeQuery(query1);

rs.next();

if(rs.first()==true)

{

JOptionPane.showMessageDialog(null,"Your account already exist");

}

else

{

String query="insert into loginfo values('"+uname+"','"+pwd+"','no','"+faltu+"','off')";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(null,"Your changes have been saved successfully!");

new admindetails().setVisible(true);

this.dispose();

}

}

else

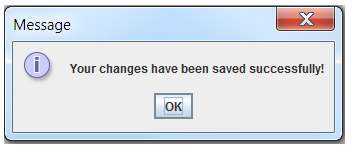
JOptionPane.showMessageDialog(null,"Empty field, please insert values.");

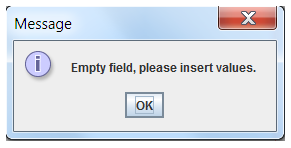
}

catch(Exception e)

{JOptionPane.showMessageDialog(null,e);}

}



****

**Code for ActionPerformed event of jButton2:-**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();

new admindetails().setVisible(true);

}

**Code for WindowActivated event :-**

private void formWindowActivated(java.awt.event.WindowEvent evt) {

try{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query="select max(id) as 'v' from loginfo";

ResultSet rs=stmt.executeQuery(query);

rs.next();

int max = rs.getInt("v");

int q = max+1;

jTextField1.setText(""+q);

}

catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

**Code for FocusLost event of jTextField2:-**

private void jTextField2FocusLost(java.awt.event.FocusEvent evt)

{

String s=jTextField2.getText();

if(s.length()!=0)

{

boolean flag=false;

for(int i=0;i<=s.length()-1;i++){

if(((s.charAt(i)>=65&&s.charAt(i)<=90)||(s.charAt(i)>=97 && s.charAt(i)<=122)||s.charAt(i)==32)==false)

{

flag=false;

break;}

else {flag=true;

}

}

if(flag==false)

{

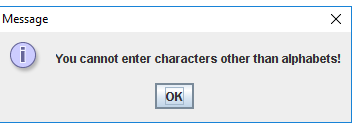
JOptionPane.showMessageDialog(null,"You cannot enter characters other than alphabets!");

jTextField2.setText("");

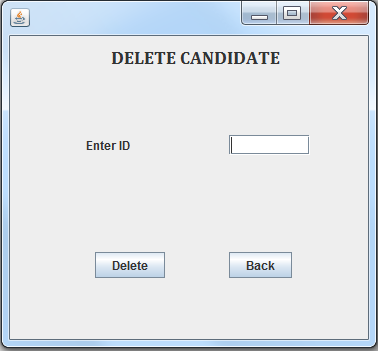
}

}

}



1. FORM TO DELETE A CANDIDATE



**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jLabel | 2 | jLabel1  jLabel2 | DELETE CANDIDATE  Enter ID |
| jTextField | 1 | jTextField3 | Nil |
| jButton | 2 | jButton1  jButton2 | Delete  Back |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

**Code for ActionPerformed event of jButton1:-**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

{

try{

String idd=jTextField3.getText();

if((idd.length()!=0))

{

int id = Integer.parseInt(idd);

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query="delete from vote where id='"+id+"'";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(null,"Your changes have been saved successfully!");

new admindetails().setVisible(true);

this.dispose();

}

else

JOptionPane.showMessageDialog(null,"Empty field, please insert values.");

}

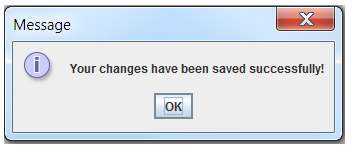
catch(Exception e)

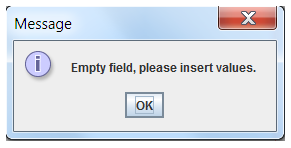
{

JOptionPane.showMessageDialog(null,e);

}

}



****

**Code for ActionPerformed event of jButton2:-**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)

{

this.dispose();

new admindetails().setVisible(true);

}

**Code for FocusLost event of jTextField3:-**

private void jTextField3FocusLost(java.awt.event.FocusEvent evt)

{

String s=jTextField3.getText();

if(s.length()!=0)

{

boolean flag=false;

for(int i=0;i<=s.length()-1;i++)

{

if((s.charAt(i)>=48&&s.charAt(i)<=57)==false)

{

flag=false;

break;

}

else{

flag=true;

}

}

if(flag==false)

{

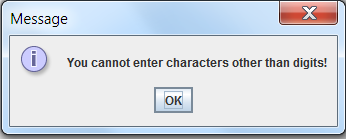
JOptionPane.showMessageDialog(null,"You cannot enter characters other than digits!");

jTextField3.setText("");

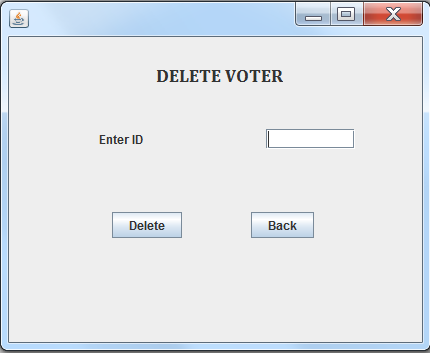
}

}

}



1. FORM TO DELETE A VOTER

****

**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jLabel | 2 | jLabel1  jLabel2 | DELETE VOTER  Enter ID |
| jTextField | 1 | JTextField1 | Nil |
| jButton | 2 | jButton1  jButton2 | Delete  Back |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

**Code for ActionPerformed event of jButton1:-**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

{

try{

String id= jTextField1.getText();

if((id.length()!=0))

{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query="delete from loginfo where id='"+id+"'";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(null,"Your changes have been saved successfully!");

new admindetails().setVisible(true);

this.dispose();

}

else

{

JOptionPane.showMessageDialog(null,"Empty field, please insert values.");

}

}

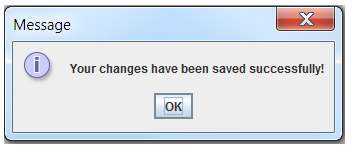
catch(Exception e)

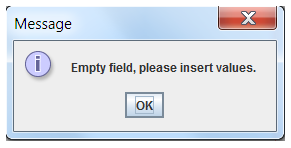
{

JOptionPane.showMessageDialog(null,e);

}

}



****

**Code for FocusLost event of jTextField1:-**

private void jTextField1FocusLost(java.awt.event.FocusEvent evt)

{

String s=jTextField1.getText();

if(s.length()!=0)

{

boolean flag=false;

for(int i=0;i<=s.length()-1;i++){

if((s.charAt(i)>=48&&s.charAt(i)<=57)==false)

{

flag=false;

break;

}

else{

flag=true;

}

}

if(flag==false)

{

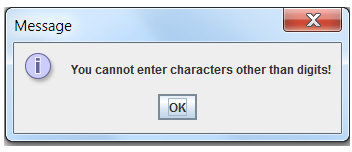
JOptionPane.showMessageDialog(null,"You cannot enter characters other than digits!");

jTextField1.setText("");

}

}

}



**Code for ActionPerformed event of jButton2:-**

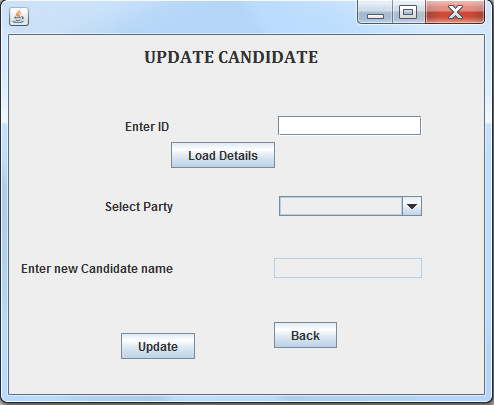
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();

new admindetails().setVisible(true);

}

1. FORM TO UPDATE A CANDIDATE



**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jLabel | 4 | jLabel1  jLabel2  jLabel3  jLabel4 | UPDATE CANDIDATE  Enter ID  Select Party  Enter new Candidate name |
| jTextField | 2 | jTextField1  jTextField2 | Nil  Nil |
| jComboBox | 1 | jComboBox1 | Nil |
| jButton | 3 | jButton1  jButton2  jButton3 | Update  Back  Load Details |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

**Code for ActionPerformed event of jButton1:-**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

{

try{

Object pname = jComboBox1.getSelectedItem();

String candidate= jTextField2.getText();

String idd=jTextField1.getText();

int id = Integer.parseInt(idd);

String query1 = "select \* from vote where id ='"+id+"'";

if((candidate.length()!=0))

{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

ResultSet rs=stmt.executeQuery(query1);

rs.next();

if(rs.first()==true)

{

String query=" update vote set party\_name ='"+pname+"',candidate\_name ='"+candidate+"'where id='"+id+"'";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(null,"Your changes have been saved successfully!");

new admindetails().setVisible(true);

this.dispose();

}

else {

JOptionPane.showMessageDialog(null,"Please enter new candidate name.");

jTextField2.setText("");

}

}

else

JOptionPane.showMessageDialog(null,"Empty field, please insert values.");

}

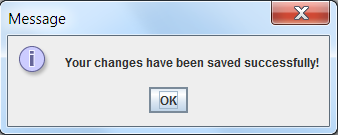
catch(Exception e)

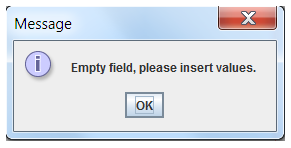
{

JOptionPane.showMessageDialog(null,e);

}

}





**Code for FocusLost event of jTextField1:-**

private void jTextField1FocusLost(java.awt.event.FocusEvent evt)

{

String s=jTextField1.getText();

if(s.length()!=0)

{

boolean flag=false;

for(int i=0;i<=s.length()-1;i++)

{

if((s.charAt(i)>=48&&s.charAt(i)<=57)==false)

{

flag=false;

break;

}

else{

flag=true;

}

}

if(flag==false)

{

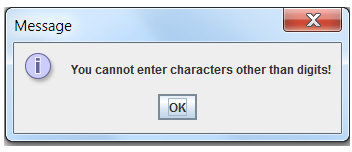
JOptionPane.showMessageDialog(null,"You cannot enter characters other than digits!");

jTextField1.setText("");

}

}

}



**Code for FocusLost event of jTextField2:-**

private void jTextField2FocusLost(java.awt.event.FocusEvent evt)

{

String s=jTextField2.getText();

if(s.length()!=0)

{

boolean flag=false;

for(int i=0;i<=s.length()-1;i++)

{

if(((s.charAt(i)>=65&&s.charAt(i)<=90)||(s.charAt(i)>=97 && s.charAt(i)<=122)||s.charAt(i)==32)==false)

{

flag=false;

break;

}

else {

flag=true;

}

}

if(flag==false)

{

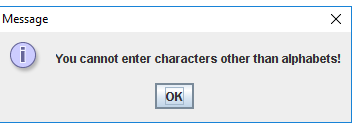
JOptionPane.showMessageDialog(null,"You cannot enter characters other than alphabets!");

jTextField2.setText("");

}

}

}



**Code for ActionPerformed event of jButton2:-**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)

{

this.dispose();

new admindetails().setVisible(true);

}

**Code for ActionPerformed event of jButton3:-**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)

{

int id = Integer.parseInt(jTextField1.getText());

try{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query1="select candidate\_name from vote where id = '"+id+"'";

ResultSet rs=stmt.executeQuery(query1);

rs.next();

String c = rs.getString("candidate\_name");

jTextField2.setText(c);

jTextField2.setEditable(true);

jTextField1.setEditable(false);

Statement stmt1=conn.createStatement();

String query="select distinct(party\_name) as 'p' from vote";

ResultSet rs1=stmt1.executeQuery(query);

while(rs1.next())

{

String pname = rs1.getString("p");

jComboBox1.addItem(pname);

}

}

catch(Exception e)

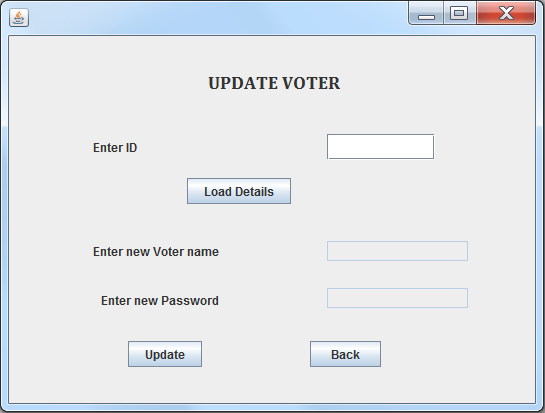
{

JOptionPane.showMessageDialog(null,e);

}

}

1. . FORM TO UPDATE A VOTER

****

**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jLabel | 4 | jLabel1  jLabel2  jLabel3  jLabel4 | UPDATE VOTER  Enter ID  Enter new Voter name  Enter new Password |
| jTextField | 3 | jTextField1  jTextField2  jTextField3 | Nil  Nil  Nil |
| jButton | 3 | jButton1  jButton2  jButton3 | Update  Back  Load Details |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

**Code for ActionPerformed event of jButton1:-**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

{

try{

String uname= jTextField2.getText();

String pwd= jTextField3.getText();

String idd=jTextField1.getText();

int id = Integer.parseInt(idd);

String query1="select \* from loginfo where id ='"+id+"'";

if((uname.length()!=0)&&(pwd.length()!=0)&&(idd.length()!=0))

{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

ResultSet rs=stmt.executeQuery(query1);

rs.next();

if(rs.first()==true)

{

String query=" update loginfo set username ='"+uname+"',password='"+pwd+"'where id='"+id+"'";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(null,"Your changes have been saved successfully!");

new admindetails().setVisible(true);

this.dispose();

}

else

{

JOptionPane.showMessageDialog(null,"Sorry,No such ID found.");

jTextField1.setText("");

jTextField2.setText("");

jTextField2.setEditable(false);

jTextField3.setText("");

jTextField3.setEditable(false);

}

}

else

{

JOptionPane.showMessageDialog(null,"Empty field, please insert values.");

}

}

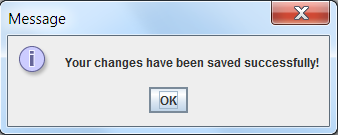
catch(Exception e)

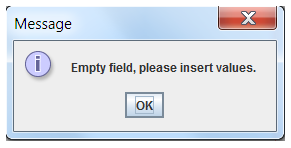
{

JOptionPane.showMessageDialog(null,e);

}

}





**Code for FocusLost event of jTextField1:-**

private void jTextField1FocusLost(java.awt.event.FocusEvent evt)

{

String s=jTextField1.getText();

if(s.length()!=0)

{

boolean flag=false;

for(int i=0;i<=s.length()-1;i++)

{

if((s.charAt(i)>=48&&s.charAt(i)<=57)==false)

{

flag=false;

break;

}

else{

flag=true;

}

}

if(flag==false)

{

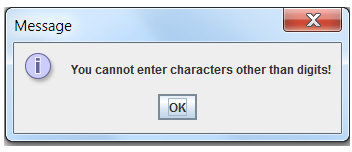
JOptionPane.showMessageDialog(null,"You cannot enter characters other than digits!");

jTextField1.setText("");

}

}

}



**Code for ActionPerformed event of jButton2:-**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)

{

this.dispose();

new admindetails().setVisible(true);

}

**Code for ActionPerformed event of jButton3:-**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)

{

int id = Integer.parseInt(jTextField1.getText());

try{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query1="select username,password from loginfo where id = '"+id+"'";

ResultSet rs=stmt.executeQuery(query1);

rs.next();

String u = rs.getString("username");

String p = rs.getString("password");

jTextField2.setText(u);

jTextField3.setText(p);

jTextField2.setEditable(true);

jTextField3.setEditable(true);

jTextField1.setEditable(false);

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,e);

}

}

**Code for FocusLost event of jTextField2:-**

private void jTextField2FocusLost(java.awt.event.FocusEvent evt)

{

String s=jTextField2.getText();

if(s.length()!=0)

{

boolean flag=false;

for(int i=0;i<=s.length()-1;i++)

{

if(((s.charAt(i)>=65&&s.charAt(i)<=90)||(s.charAt(i)>=97 && s.charAt(i)<=122)||s.charAt(i)==32)==false)

{

flag=false;

break;

}

else {

flag=true;

}

}

if(flag==false)

{

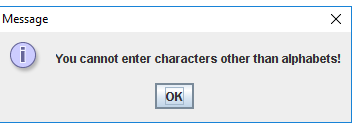
JOptionPane.showMessageDialog(null,"You cannot enter characters other than alphabets!");

jTextField2.setText("");

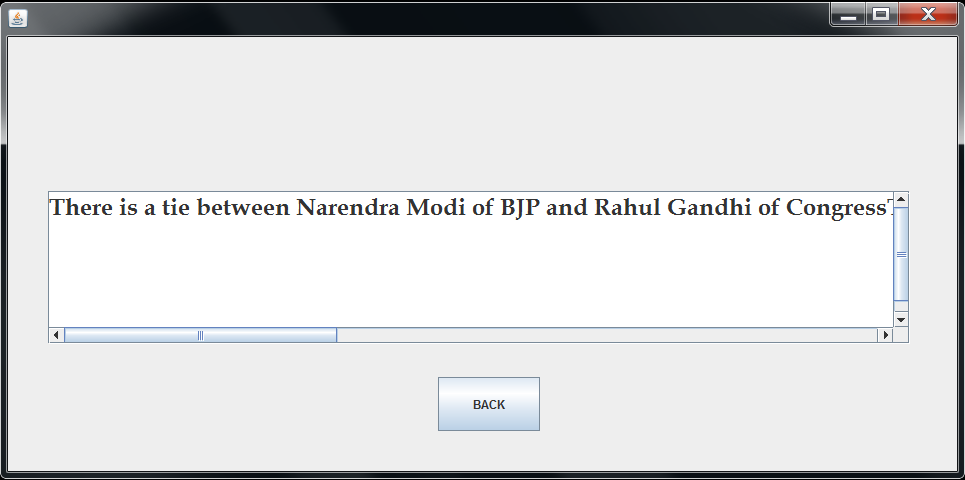
}

}

}



1. RESULT



**DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| *Component used* | *No. of component* | *Name of variables* | *Text in component* |
| jTextArea | 1 | jTextArea1 | Nil |
| jButton | 1 | jButton1 | BACK |

**Code for importing pre-requisite files:-**

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

**Code for WindowActivated event :-**

private void formWindowActivated(java.awt.event.WindowEvent evt) {

setExtendedState(6);

try{

Class.forName("java.sql.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/candidate","root","");

Statement stmt=conn.createStatement();

String query1="Select t1.p1 as winparty ,t1.sv1 as maxvotes from (select party\_name as p1,sum(votes) as sv1 from vote group by party\_name ) as t1 join (select max(t3.sv1) as msv2 from (select party\_name as p1,sum(votes) as sv1 from vote group by party\_name ) as t3 ) as t2 on (t1.sv1=t2.msv2) ";

ResultSet rs=stmt.executeQuery(query1);

int count=0;

while(rs.next())

{

count++;

}

rs.first();

if(count>1){

jTextArea1.setText("There is a condition of tie!");

return;

}

String pname=rs.getString("winparty");

String votes=rs.getString("maxvotes");

Statement stmt1=conn.createStatement();

String query2="select tt1.candidate\_name as wincandidate,tt1.mv1 as winvotes from (select candidate\_name,max(votes) as mv1 from vote where party\_name =(select t1.p1 as winp from (select party\_name as p1,sum(votes) as sv1 from vote group by party\_name ) as t1 join (select max(t3.sv1) as msv2 from (select party\_name as p1,sum(votes) as sv1 from vote group by party\_name) as t3 ) as t2 on (t1.sv1=t2.msv2)) group by candidate\_name) as tt1 join (select max(votes) as mv2 from vote where party\_name =(select t1.p1 from (select party\_name as p1,sum(votes) as sv1 from vote group by party\_name ) as t1 join (select max(t3.sv1) as msv2 from (select party\_name as p1 ,sum(votes) as sv1 from vote group by party\_name) as t3) as t2 on (t1.sv1=t2.msv2))) as tt2 on (tt1.mv1=tt2.mv2)";

ResultSet rs1=stmt.executeQuery(query2);

rs1.next();

String cname=rs1.getString("wincandidate");

int vote=rs1.getInt("winvotes");

jTextArea1.append("The winner party is "+pname+"\n with total votes "+ votes+" and the newly elected candidate is "+ cname+"\n "+"with maximum votes "+ vote+".");

}

catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

**Code for ActionPerformed event of jButton1:-**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

{

this.dispose();

new admindetails().setVisible(true);

}

***Database in MySQL***

Database name: candidate

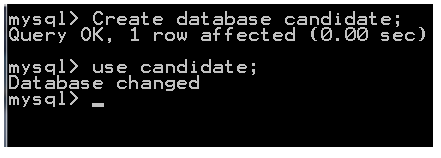
Tables in candidate are:

Admin

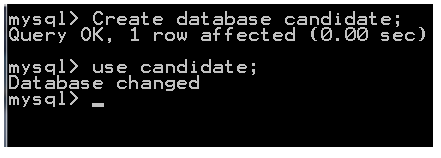
Loginfo

Vote

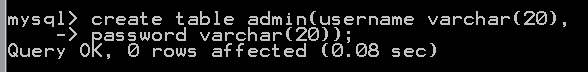
Creating database in MySQL



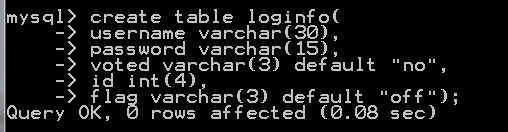
Using the above created database



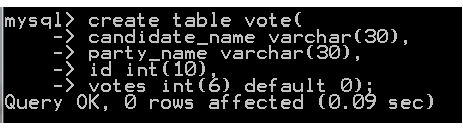
Creating table admin



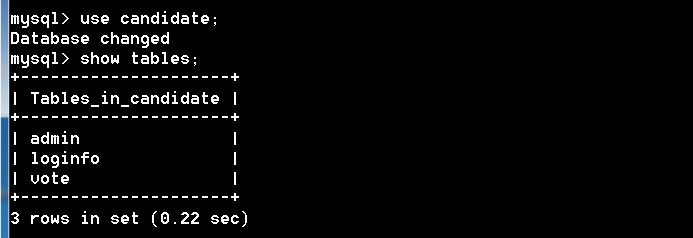
Creating table loginfo



Creating table vote



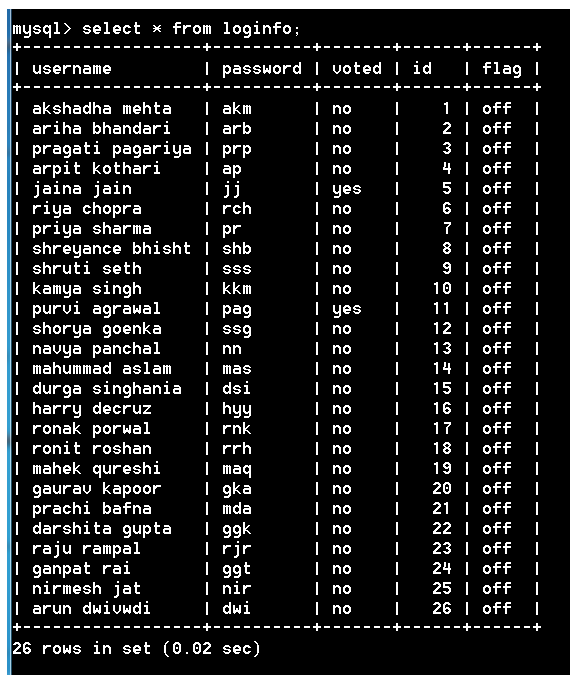
Tables in database



Example of data in admin table



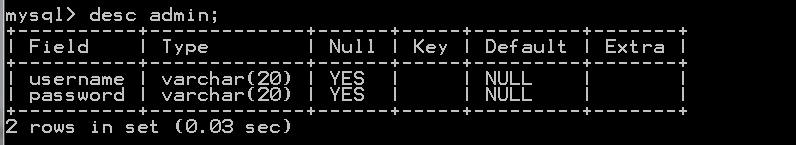
Example of data in loginfo table



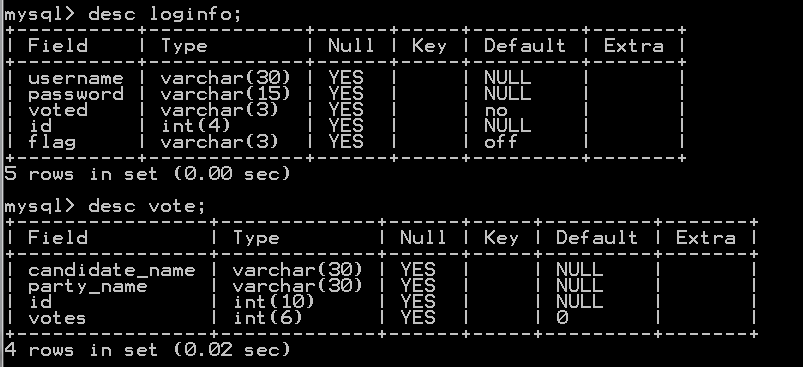
Example of data in vote table



Desc command on admin table



Desc command on loginfo table



Desc command on vote table

