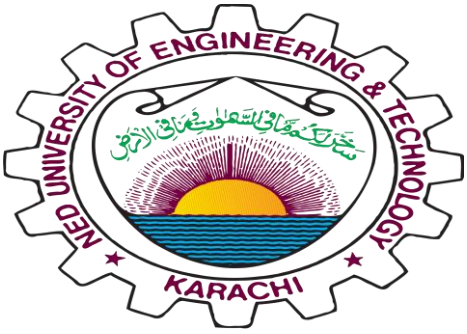
# THE ELECTRONIC VOTING SYSTEM

**DEVELOPED BY:**
**WAQAR GUL (ROLL NO.: SE-37, SECTION A)**
**MASOOD UR REHMAN (ROLL NO.: SE-09, SECTION A)**

# PROJECT REPORT


**SUBMITTED TO:**
**MS. ASMA KHAN**
**SUBJECT: OBJECT ORIENTED PROGRAMMING**
**DISCIPLINE: SOFTWARE ENGINEERING**

# TABLE OF CONTENTS

# INTRODUCTION

## PROBLEM DEFINITION

The existing manual Voting system consumes more time for Vote Casting. Voter has to wait in long queue to vote for a right candidate. The election officers has to be check the voter credentials manually, which is a faulty and tiring process. The voter had to stand in the queue in order to access polling booths. All the work is done in paper ballot thus, manual counting of vote leads to tempering of results and loss of votes. To overcome of all these problems we have to implement a web application, which is helpful for Voting from anywhere.


## OBJECTIVES

The objective of the system is a replacement of the traditional system that is in existence. This smart system reduces the time for voting and also the system is reliable, and faster. In this system the voter OTP will be sent through email registered in database. Database maintained by this system usually contains the Voters information, Candidate information and the final results of total elections.

# SYSTEM REQUIREMENTS

The basic components required in developing and using the system are listed below.

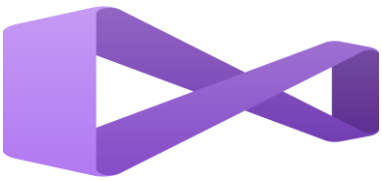
## 1. HARDWARE REQUIREMENTS

- Processor: Pentium
- RAM: 4GB
- Hard Disk: 1TB
- Speed: 1.1GHz

## 2. SOFTWARE REQUIREMENTS

- Database Management System: MYSQL
- Front-End Development: Microsoft Visual Studio
- Programming Language: C++/C#
- Libraries: Mail Client, SQL Client & Drawing



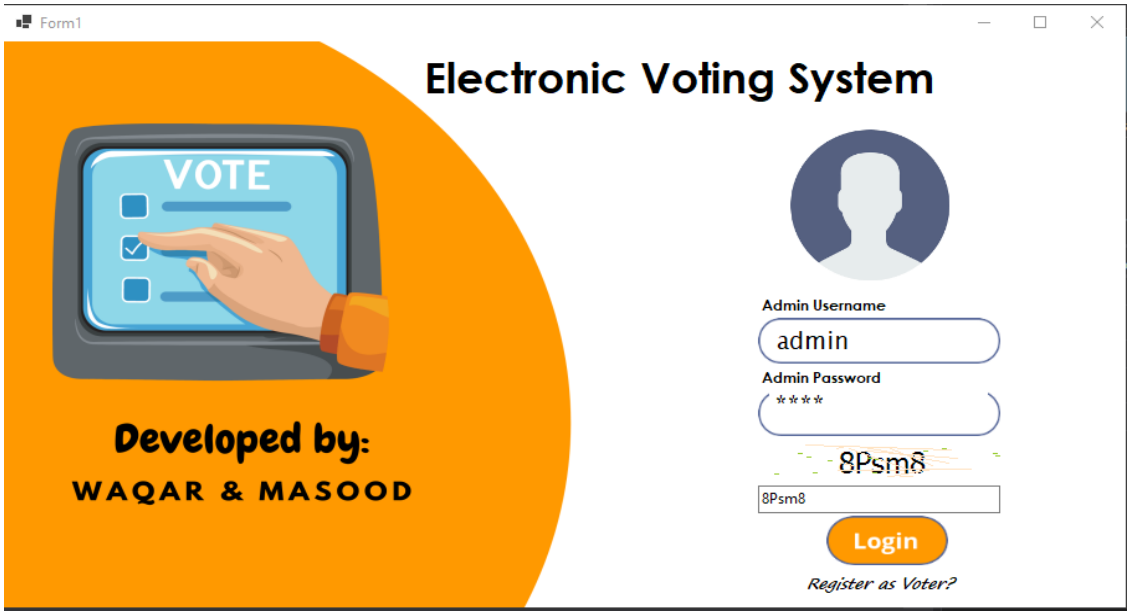


# MODULES OF PROPOSED SYSTEM

This system is designed keeping administration control and voter usage along with essential restrictions, validations and verifications. The system can be divided into three modules that are discussed below:

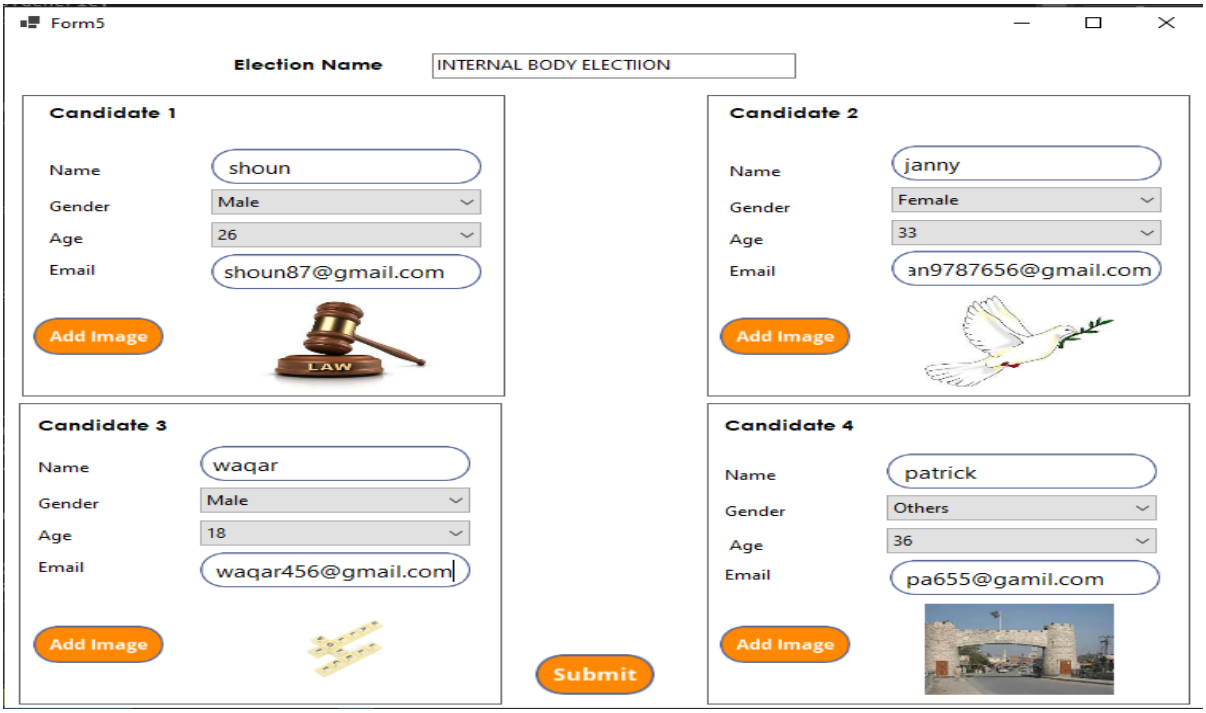
1. **ADMINISTRATIVE MODULE:**

   This module is mainly concerned with the administrative controls provided to the governing authority. This allows the administrator to manage and monitor the election process. It allows the admin to login into the system by entering “Username & Password” provided by manufacturer. These credentials are then verified and validated to avoid misuse of system. Furthermore, to assure human access “Re-Captcha” is used so that no machine can access the system. As shown in picture below:

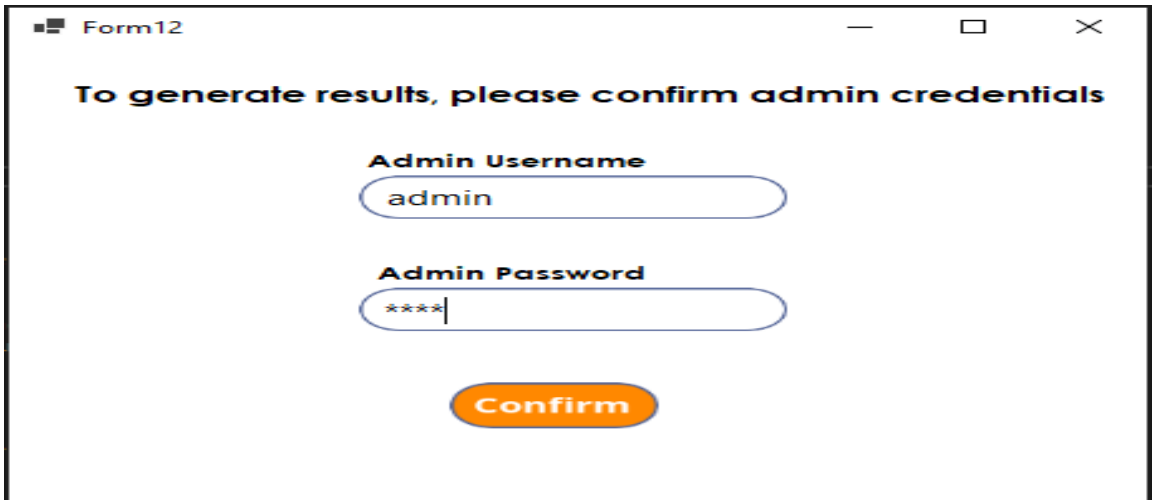


Once admin is allowed the access, he/she is allowed to create, monitor, stop or generate the results of election. As show below:


If the admin wants to create the election, then nominee details have to be entered that include fields of name, age, gender, email and voting sign. All these fields are validated upon entry to avoid fraud or wrong information input, after which the details are stored in the database table:



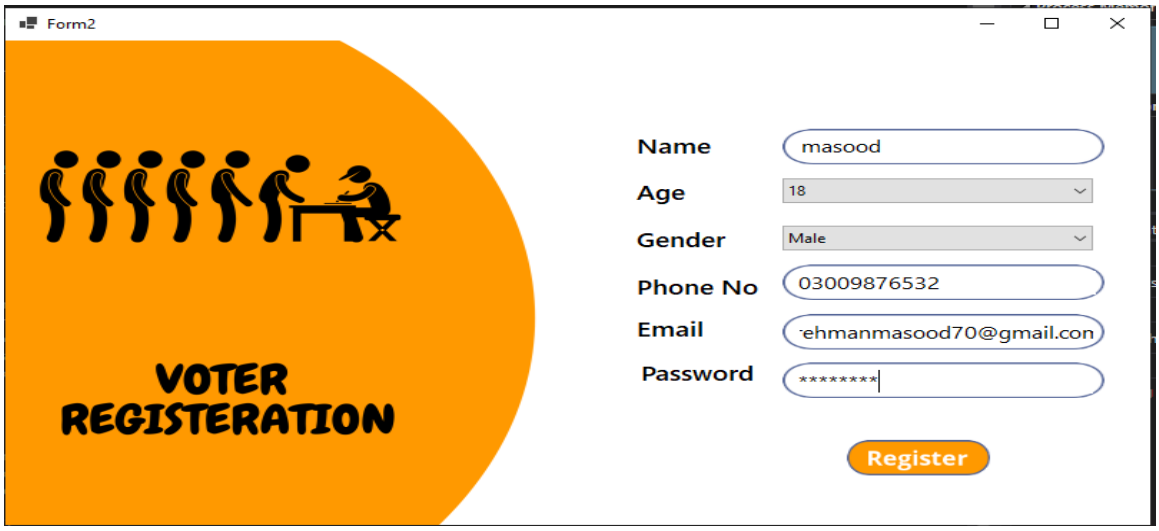
Upon generation of result, the winner of the election will be shown along with his information stored in database and the votes gained by the candidate:



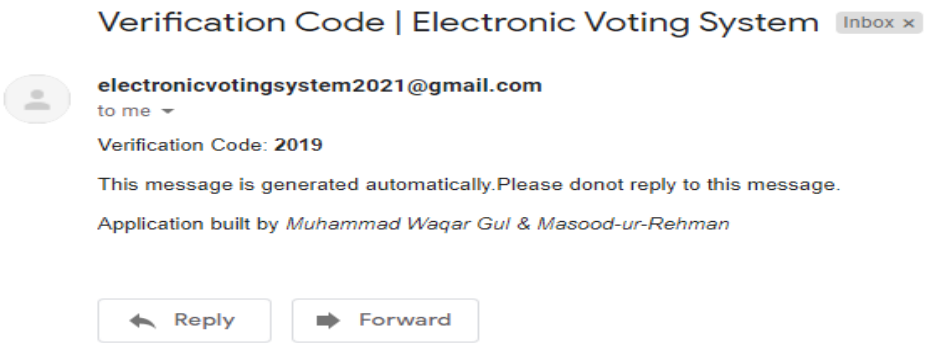

## 2. VOTER/USER MODULE:

This section deals with the functionality provided to the voters under the strict validation and verification rules. The voters have to first register into the system by entering some basic personal information and an email and password to access the system later. The email id is validated to check correct input and thus to verify an OTP is mailed. If the entered OTP matches, then the voter will be registered.

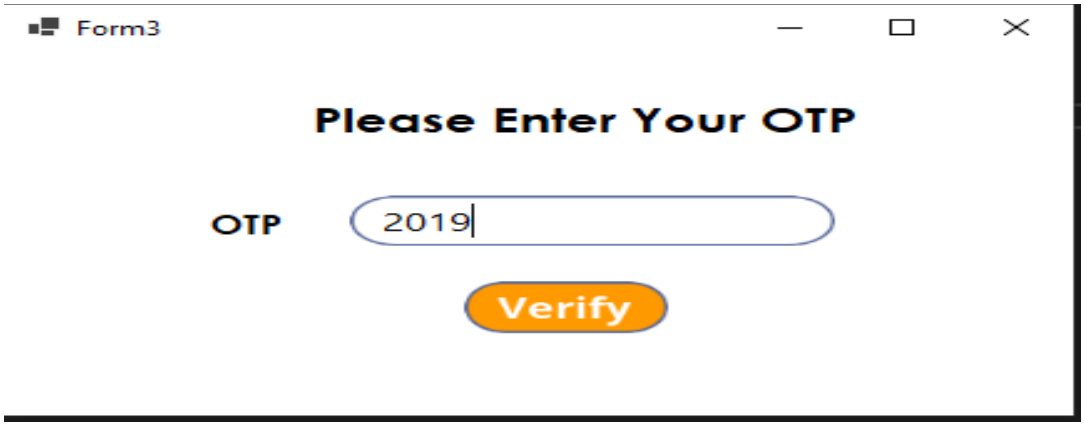
The data entered by voter will be stored in database for future reference. This also allow single vote per user as duplicative registration is impossible:

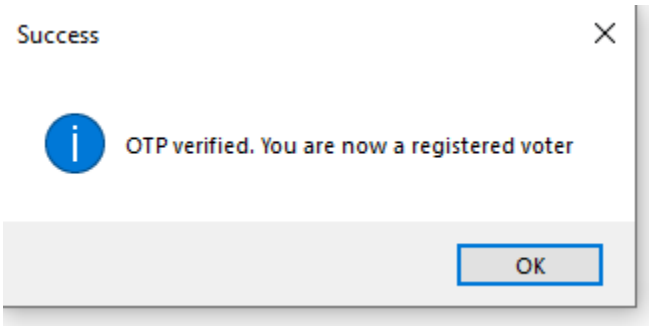


Below is the screenshot of email that will be received.

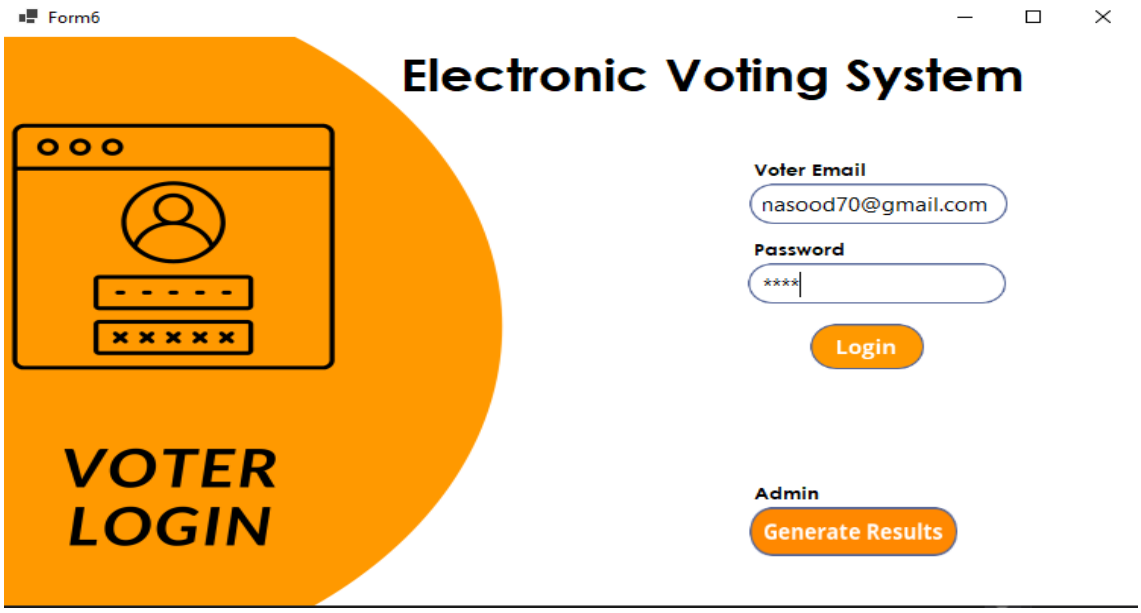



As the OTP is verified, a message box will be displayed notifying the voter.

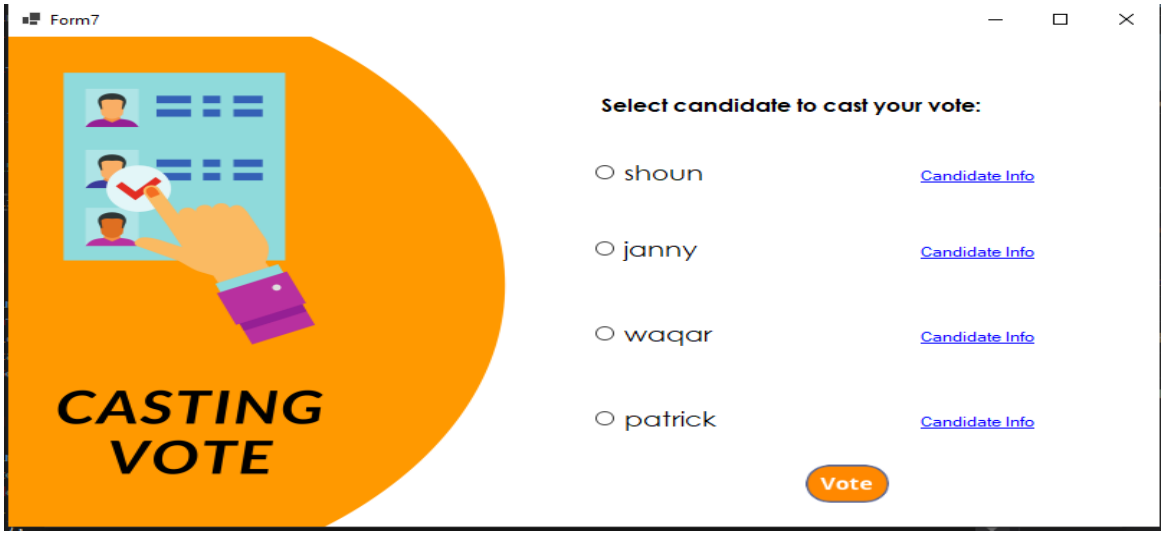


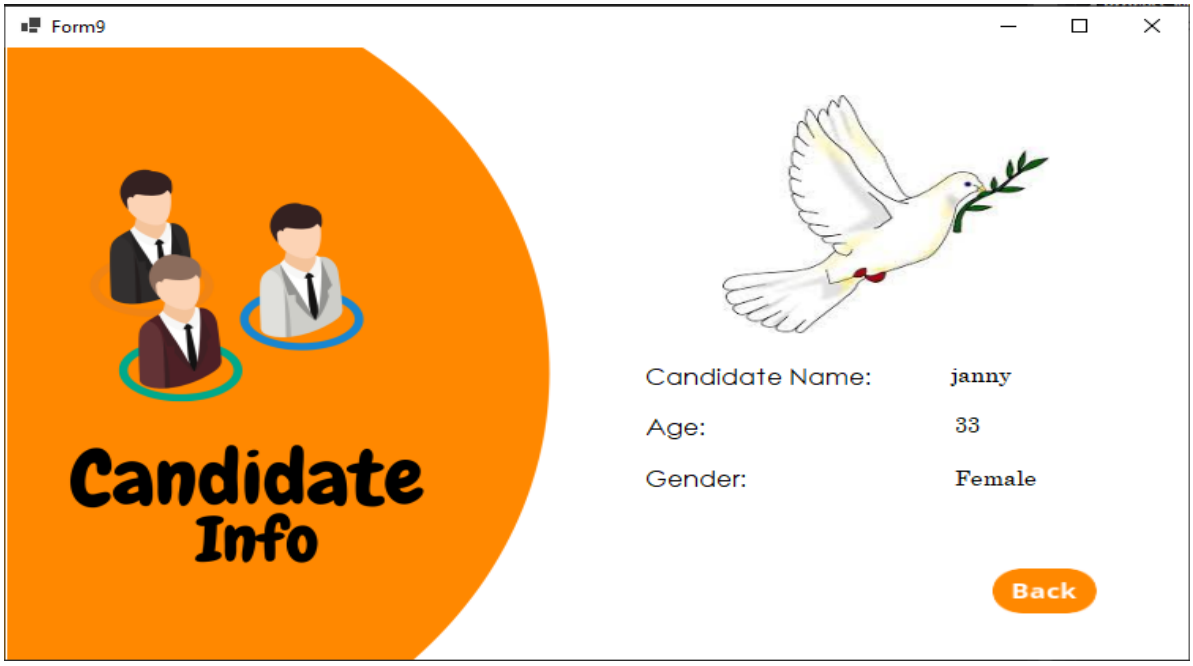
In order to vote, the voter must enter same email address and password with which they registered. This will be checked against the data in database to avoid misuse of right to vote:



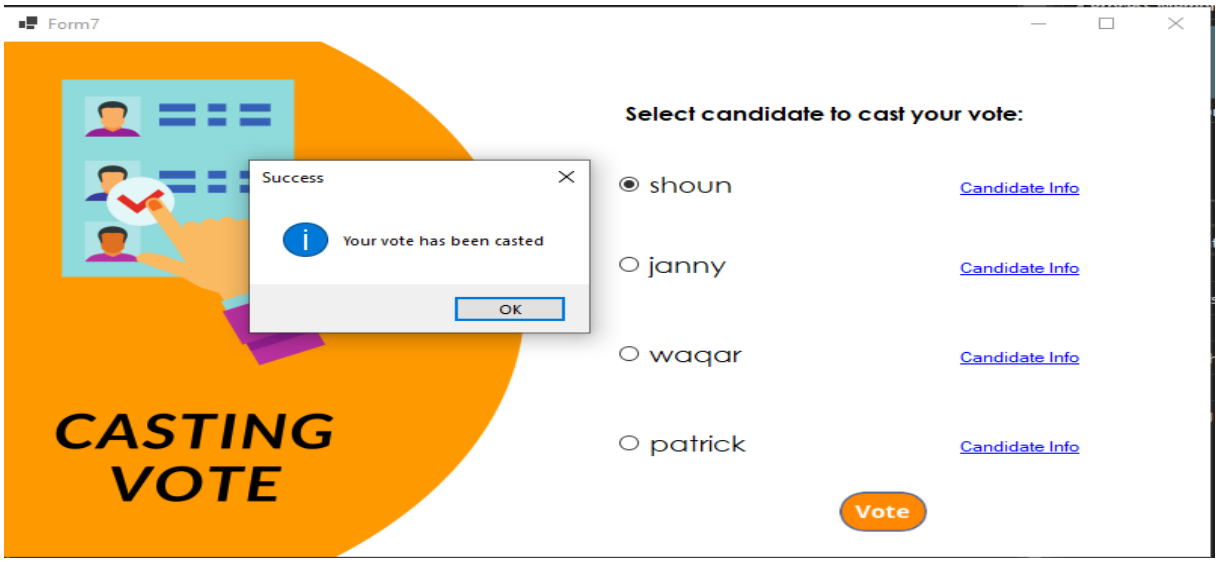

If the credentials matched, then the user will be allowed to select any one Candidate at a time. And the user can access the candidate information via “Info” button to help in making decision. The candidate info will be retrieved from the database in which data was stored by admin:



Below are some sample snippets of the output, if the voter clicks on the "Candidate Info" link on ballot window:

Voter is restricted to vote single candidate by the use of “Radio Buttons”. Furthermore, once the user have confirmed their selection by clicking “Vote” button, then the Ballot window will be freeze and the message box will be displayed as confirmation.

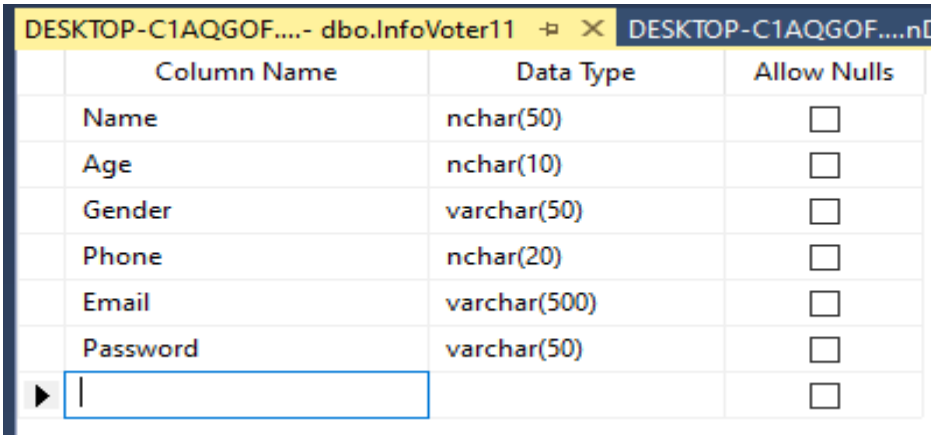


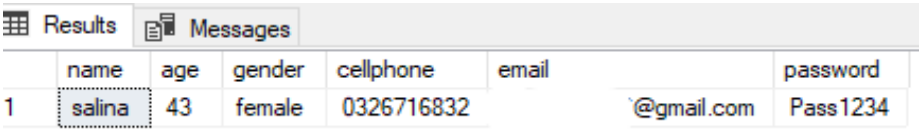
## 3. DATABASE MODULE:

This involves the tables created in MySQL to store the data of the system. It is distributed into four parts as shown below:

- *Voters Table*:

  This table maintains the voter records that the voter enters while registration. It is then used to verify voter's data during voter login process to avoid fraud. It is mainly linked with the Voter Module of the system.



- *Candidate Table*:

  This table maintains the candidate records that the admin enters while registration. It is then used to retrieve candidate's data during balloting process to avoid fraud. It is mainly linked with the Voter Module and Administrative Module of the system.

DESKTOP-C1AQGOF....B - dbo.Candidate* | SQLQuery1.sql - D...QG

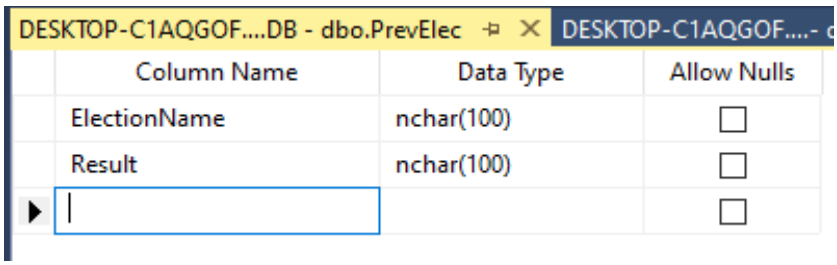
| Column Name | Data Type | Allow Nulls |
|---|---|---|
| Name | nchar(30) | ☐ |
| Gender | nchar(10) | ☐ |
| Age | nchar(10) | ☐ |
| Sno | nchar(10) | ☐ |
| Email | nchar(100) | ☐ |
| | | ☐ |

Results | Messages

| name | age | gender | email | image |
|---|---|---|---|---|
| shahida | 75 | female | shahida65@gmail.com | 0x53797374656D2E44726177696E672E4269746D6170 |
| shakoor | 32 | male | shakoor63@gmail.com | 0x53797374656D2E44726177696E672E4269746D6170 |
| sohaib | 25 | male | sohaib98@gmail.com | 0x53797374656D2E44726177696E672E4269746D6170 |
| shakoor | 32 | male | shakoor87@gmail.com | 0x53797374656D2E44726177696E672E4269746D6170 |

- *Previous Election*:

  This table hold and maintains the results of the previous election made using the system. It can be used in future to recheck the results of any election. It is mainly linked with the Administrative Module of the system.




- *Results*:

  There is no separate table for this section. However, the results will be stored in table of “Previous Elections”. It is used to declare the election results. It is mainly linked with the Administrative Module of the system.

# CODE SNIPPETS

## ADMIN CREDENTIAVERIFICATION:

```
if (textBox1.Text.ToString() != "admin" || textBox2.Text.ToString() != "a123")
{
    MessageBox.Show("Incorrect credentials", "Wrong Credentials", MessageBoxButtons.OK, MessageBoxIcon.Error)

}

else
{
    if (textBox3.Text == finalString)
    {

        this.Hide();
        Form4 f4 = new Form4();
        f4.Show();

    }
    else if(textBox3.Text != finalString || textBox3.Text.TrimStart() == "")
    {
        MessageBox.Show("Enter valid captcha", "Invalid Captcha", MessageBoxButtons.OK, MessageBoxIcon.Error)
        finalString = " ";
        CaptchaString();
        pictureBox6.Image = CaptchaToImage(finalString, pictureBox2.Width, pictureBox2.Height);
    }
```

## CAPTCHA CREATOR:

```
private string CaptchaString()
{
    var chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
    var stringChars = new char[5];
    var random = new Random();

    for (int i = 0; i < stringChars.Length; i++)
    {
        stringChars[i] = chars[random.Next(chars.Length)];
    }

    finalString = new String(stringChars);
    return finalString;
}

2 references
private Bitmap CaptchaToImage(string text, int width, int height)
{
    Bitmap bmp = new Bitmap(width, height);
    Graphics g = Graphics.FromImage(bmp);
    SolidBrush sb = new SolidBrush(Color.White);
    g.FillRectangle(sb, 0, 0, bmp.Width, bmp.Height);
    Font font = new Font("Tahoma", 18);
    sb = new SolidBrush(Color.Black);
    g.DrawString(text, font, sb, bmp.Width / 2 - (text.Length / 2) * font.Size, (bmp.Height / 2) - font.Size);
    int count = 0;
    Random rand = new Random();
    while (count < 20)
    {
        sb = new SolidBrush(Color.YellowGreen);
        g.FillEllipse(sb, rand.Next(0, bmp.Width), rand.Next(0, bmp.Height), 4, 2);
        count++;
    }
    count = 0;
    while (count < 10)
    {
        g.DrawLine(new Pen(Color.Bisque), rand.Next(0, bmp.Width), rand.Next(0, bmp.Height), rand.Next(0, bmp.Wid
        count++;
    }
    return bmp;
}
```

## VOTER REGISTRATION (VALIDATIONS AND DATABASE STORAGE):

```
1 reference
private void pictureBox5_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(conString);
    conn.Open();
    string q = "SELECT Email FROM InfoVoter11 WHERE Email ='" + textBox4.Text.ToString() + "'";
    SqlCommand cmd = new SqlCommand(q, conn);

    string iemail = cmd.ExecuteScalar() as string;

    Regex mRegxExpression;
    Regex mobilepattern;
    mobilepattern = new Regex(@"^[0-9]{11}$");
    if (textBox1.Text.Trim() == string.Empty || textBox2.Text.Trim() == string.Empty || textBox3.Text.Trim() == strin
        textBox4.Text.Trim() == string.Empty || comboBox1.Text.Trim() == string.Empty || comboBox2.Text.Trim() ==
    {
        MessageBox.Show("All fields are required","Error",MessageBoxButtons.OK,MessageBoxIcon.Error);

    }
    else if(!mobilepattern.IsMatch(textBox2.Text.Trim()))
    {
        MessageBox.Show("Phone number enterd is incorrect.", "Incorrect Format", MessageBoxButtons.OK, MessageBoxIcon
        textBox2.Focus();
    }
    else
    {
        mRegxExpression= new Regex(@"cloud\.neduet\.edu\.pk");

        if (!mRegxExpression.IsMatch(textBox4.Text.Trim()))
        {
            MessageBox.Show("E-mail address format is not correct.", "Incorrect Format", MessageBoxButtons.OK, Messag
            textBox4.Focus();
        }
        else if (iemail == textBox4.Text.ToString())
        {
            MessageBox.Show("You are already a registered voter","Registered Already",MessageBoxButtons.OK,MessageBox
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();


        }


        else
        {
            sendmail();
        }
```

```
0 references
public void sql()
{
    SqlConnection conn = new SqlConnection(conString);
    conn.Open();
    if (conn.State == System.Data.ConnectionState.Open)
    {
        string q = "insert into InfoVoter11(Name,Age,Gender,Phone,Email,Password)values('" + textBox1.Text.ToString()
            comboBox1.Text.ToString() + "','"+comboBox2.Text.ToString()+ "','"+textBox2.Text.ToString()+ "','"+
            textBox4.Text.ToString()+ "','"+ textBox3.Text.ToString()+"')";
        SqlCommand cmd = new SqlCommand(q, conn);
        cmd.ExecuteNonQuery();


    }

}
```

## EMAIL & OTP GENERATOR:

```
1 reference
public void sendmail()
{
    Random rnd = new Random();
    otp = rnd.Next(1000, 9999);
    try
    {

        MailMessage message = new MailMessage();

        SmtpClient smtp = new SmtpClient();

        message.IsBodyHtml = true;
        message.From = new MailAddress("electronicvotingsystem2021@gmail.com");

        message.To.Add(new MailAddress(textBox4.Text.ToString()));

        message.Subject = "Verification Code | Electronic Voting System";

        message.Body = @"<head> Verification Code: " + "<b>" + otp+"</b>"+"</head>" +
            "<p>" +
            "" +
            "This message is generated automatically.Please donot reply to this message.</ p > " +
            "<p>Application built by <i> Muhammad Waqar Gul & Masood-ur-Rehman </i></p>    ";

        smtp.Port = 587;

        smtp.Host = "smtp.gmail.com";
        smtp.EnableSsl = true;
        smtp.UseDefaultCredentials = false;
        smtp.Credentials = new NetworkCredential("electronicvotingsystem2021@gmail.com", "electronicvoting2021");
        smtp.DeliveryMethod = SmtpDeliveryMethod.Network;

        smtp.Send(message);
        MessageBox.Show("4 digit OTP has been send to your email. Please verify to get registered as voter.","Success
        name = textBox1.Text.ToString();
        age = comboBox1.Text.ToString();
        gender = comboBox2.Text.ToString();
        phone = textBox2.Text.ToString();
        email = textBox4.Text.ToString();
        password = textBox3.Text.ToString();
        Form3 f3 = new Form3();
        f3.Show();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("err: " + ex.Message,"Error",MessageBoxButtons.OK,MessageBoxIcon.Error);
    }
```

## DELETE VOTER:

```
1 reference
private void pictureBox4_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(conString);
    conn.Open();
    string q = "DELETE FROM InfoVoter11";
    SqlCommand cmd = new SqlCommand(q, conn);
    cmd.ExecuteNonQuery();
    MessageBox.Show("Database Updated.", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
    conn.Close();

}
```

## IMAGE UPLOAD FOR CANDIDATE:

```
1 reference
private void pictureBox23_Click(object sender, EventArgs e)
{
    OpenFileDialog opnfd = new OpenFileDialog();
    opnfd.Filter = "Image Files (*.jpg;*.jpeg;.*.gif;)|*.jpg;*.jpeg;.*.gif";
    if (opnfd.ShowDialog() == DialogResult.OK)
    {
        pictureBox3.Image = new Bitmap(opnfd.FileName);
        image3 = pictureBox3.Image;
    }
}
```

## STORAGE OF CANDIDATE DATA IN DATABASE:

```
1 reference
private void pictureBox25_Click(object sender, EventArgs e)
{
    if (textBox1.Text.Trim()==string.Empty){
        MessageBox.Show("Election Name cannot be empty.","Error",MessageBoxButtons.OK,MessageBoxIcon.Warning);
    }
    else {

    SqlConnection conn = new SqlConnection(conString);
    conn.Open();
    string q = "insert into Candidate(Name, Gender, Age, Email,Sno)values('" + textBox18.Text.ToString() + "', '" + c
        comboBox2.Text.ToString() + "', '" + textBox21.Text.ToString() + "', '" + 1 + "') ";
    string q1 = "insert into Candidate(Name, Gender, Age, Email,Sno)values('" + textBox22.Text.ToString() + "', '" +
      comboBox4.Text.ToString() + "', '" + textBox6.Text.ToString() + "', '" + 2 + "') ";
    string q2 = "insert into Candidate(Name, Gender, Age, Email,Sno)values('" + textBox24.Text.ToString() + "', '" +
     comboBox6.Text.ToString() + "', '" + textBox12.Text.ToString() + "', '" + 3 + "') ";
    string q3 = "insert into Candidate(Name, Gender, Age, Email,Sno)values('" + textBox15.Text.ToString() + "', '" +
      comboBox8.Text.ToString() + "', '" + textBox16.Text.ToString() + "', '" + 4 + "') ";
    string q4 = "delete from Evoters";
    SqlCommand cmd = new SqlCommand(q, conn);
    SqlCommand cmd1 = new SqlCommand(q1, conn);
    SqlCommand cmd2 = new SqlCommand(q2, conn);
    SqlCommand cmd3 = new SqlCommand(q3, conn);
    SqlCommand cmd4 = new SqlCommand(q4, conn);
    cmd.ExecuteNonQuery();
    cmd1.ExecuteNonQuery();
    cmd2.ExecuteNonQuery();
    cmd3.ExecuteNonQuery();
    cmd4.ExecuteNonQuery();
    elecname = textBox1.Text;
    Form6 f6 = new Form6();
    this.Hide();
    f6.Show();
        }
}
```

## VERIFYING VOTER ON LOGIN:

```
private void pictureBox3_Click(object sender, EventArgs e)
{

    SqlConnection conn = new SqlConnection(conString);
    conn.Open();
    SqlCommand cmd2 = new SqlCommand("Select Email from Evoters where Email= @email", conn);
    cmd2.Parameters.AddWithValue("@email", this.textBox1.Text);
    var email = cmd2.ExecuteScalar();


    SqlCommand cmd = new SqlCommand("Select Password from InfoVoter11 where Email= @email", conn);
    cmd.Parameters.AddWithValue("@email", this.textBox1.Text);
    var pswrd = cmd.ExecuteScalar();




    if (pswrd != null)
    {
        if (email is null)
        {

            if (email != textBox1.Text.ToString())
            {
                if (pswrd.ToString() == textBox2.Text.ToString())
                {
                    MessageBox.Show("Credentials Verifed", "Success", MessageBoxButtons.OK, MessageBoxIcon.Informatio
                    string q1 = "insert into Evoters(Email)values('" + textBox1.Text.ToString() + "')";
                    SqlCommand cmd1 = new SqlCommand(q1, conn);
                    cmd1.ExecuteNonQuery();

                    textBox1.Clear();
                    textBox2.Clear();
                    textBox1.Focus();

                    Form7 f7 = new Form7();
                    f7.Show();
                }
                else
                {
                    MessageBox.Show("Wrong Credentials. Please try again", "Operation Failed", MessageBoxButtons.OK,
                    textBox1.Clear();
                    textBox2.Clear();
                    textBox1.Focus();
                }
            }
        }

        else
        {
            if(email.ToString() != textBox1.Text.ToString())
            {
                MessageBox.Show("You have already casted your vote.", "Voted Already", MessageBoxButtons.OK, MessageB
                textBox1.Clear();
                textBox2.Clear();
                textBox1.Focus();
            }
        }
    }
    else
    {
        MessageBox.Show("Voter is not registered.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        textBox1.Clear();
        textBox2.Clear();
        textBox1.Focus();
    }


reference
rivate void pictureBox4_Click(object sender, EventArgs e)
```

## DATA RETRIEVAL FROM DATABASE FOR BALLOT PAPER:

```
public Form7()
{
    InitializeComponent();
    SqlConnection conn = new SqlConnection(conString);
    conn.Open();

    SqlCommand cmd = new SqlCommand("Select Name from Candidate where Sno= @sno", conn);
    cmd.Parameters.AddWithValue("@sno", 1);
    SqlCommand cmd1 = new SqlCommand("Select Name from Candidate where Sno= @sno", conn);
    cmd1.Parameters.AddWithValue("@sno", 2);
    SqlCommand cmd2 = new SqlCommand("Select Name from Candidate where Sno= @sno", conn);
    cmd2.Parameters.AddWithValue("@sno", 3);
    SqlCommand cmd3 = new SqlCommand("Select Name from Candidate where Sno= @sno", conn);
    cmd3.Parameters.AddWithValue("@sno", 4);

    var name = cmd.ExecuteScalar();
    var name1 = cmd1.ExecuteScalar();
    var name2 = cmd2.ExecuteScalar();
    var name3 = cmd3.ExecuteScalar();
    if (name.ToString().Trim()== "")
    {
        radioButton1.Enabled = false;
        linkLabel1.Visible = false;
    }
    if (name1.ToString().Trim() == "")
    {
        radioButton2.Enabled = false;
        linkLabel2.Visible = false;
    }
    if (name2.ToString().Trim() == "")
    {
        radioButton3.Enabled = false;
        linkLabel3.Visible = false;
    }
    if (name3.ToString().Trim() == "")
    {
        radioButton4.Enabled = false;
        linkLabel4.Visible = false;
    }
    radioButton1.Text = name.ToString();
    radioButton2.Text = name1.ToString();
    radioButton3.Text = name2.ToString();
    radioButton4.Text = name3.ToString();
    candname1 = name.ToString();
    candname2 = name1.ToString();
    candname3 = name2.ToString();
    candname4 = name3.ToString();

}
```

## VOTE COUNT AND VALIDATIONS:

```
1 reference
private void pictureBox1_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked == true)
    {
        cand1 += 1;

    }
    else if (radioButton2.Checked == true)
    {

        cand2 += 1;
    }
    else if (radioButton3.Checked == true)
    {

        cand3 += 1;
    }
    else if (radioButton4.Checked == true)
    {

        cand4 += 1;
    }
    else
    {
        MessageBox.Show("Please select a valid option.", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    int total = 0;
    total = cand1 + cand2 + cand3 + cand4;
    MessageBox.Show("Your vote has been casted", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);


    this.Hide();
}
```

## DATA RETRIEVAL FROM DATABASE FOR PREVIOUS ELECTION:

```
public Form14()
{
    InitializeComponent();
    SqlConnection conn = new SqlConnection(conString);
    conn.Open();
    string q = "select (ElectionName) from PrevElec";
    SqlCommand cmd = new SqlCommand(q, conn);
    SqlDataReader reader;
    try
    {


        reader = cmd.ExecuteReader();
        while (reader.Read())
        {
            string sName = reader.GetString("ElectionName");
            comboBox1.Items.Add(sName);
        }

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    conn.Close();

}

1 reference
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)...

1 reference
private void pictureBox1_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(conString);
    conn.Open();
    try
    {
        string q = "select Result FROM PrevElec WHERE ElectionName= '"+ comboBox1.Text.ToString()+"'";
        SqlCommand cmd = new SqlCommand(q, conn);
        var reslt = cmd.ExecuteScalar();
        conn.Close();
        label1.Text = reslt.ToString();
        label1.Visible = true;
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

## GENERATE RESULT AND DATABASE UPDATE:

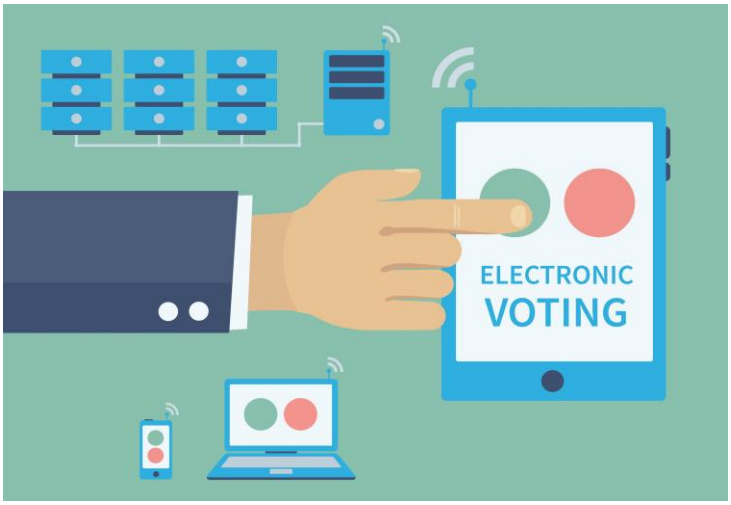
```
if (Form7.cand1> Form7.cand2 && Form7.cand1 > Form7.cand3 && Form7.cand1 > Form7.cand4)
{
    label1.Text = $"{Form7.candname1.Trim()} has won by  {Form7.cand1} votes.";
}
else if (Form7.cand2 > Form7.cand1 && Form7.cand2 > Form7.cand3 && Form7.cand2 > Form7.cand4)
{
    label1.Text = $"{Form7.candname2.Trim()} has won by  {Form7.cand2} votes.";
}
else if (Form7.cand3 > Form7.cand1 && Form7.cand3 > Form7.cand2 && Form7.cand3 > Form7.cand4)
{
    label1.Text = $"{Form7.candname3.Trim()} has won by  {Form7.cand3} votes.";
}
else if (Form7.cand4 > Form7.cand1 && Form7.cand4 > Form7.cand2 && Form7.cand4 > Form7.cand3)
{
    label1.Text = $"{Form7.candname4.Trim()} has won by  {Form7.cand4} votes.";
}
else
{
    label1.Text = "There is a tie";

}
resultdb = label1.Text;
label2.Text = $"{Form7.candname1.Trim()} secured {Form7.cand1} votes.";
label3.Text = $"{Form7.candname2.Trim()} secured {Form7.cand2} votes.";
label4.Text = $"{Form7.candname3.Trim()} secured {Form7.cand3} votes.";
label5.Text = $"{Form7.candname4.Trim()} secured {Form7.cand4} votes.";

SqlConnection conn = new SqlConnection(conString);
conn.Open();
string q = "insert into PrevElec(ElectionName, Result)values('" + Form5.elecname.ToString() + "', '" + Form13.resultc
SqlCommand cmd = new SqlCommand(q, conn);
cmd.ExecuteNonQuery();
```

# CONCLUSION

This E-Balloting system is more efficient and error less as compared to traditional manual voting system. This allows the voters to vote at anytime from anywhere thus increasing the turnout. Although, it is a small scale system designed for office, educational institutes and societies. However, this can be brought to national level with some improvisation. Thus it's time to bring some change and to listen peoples "True Voice".





***YOUR RIGHT | YOUR VOTE | YOUR DUTY***