CHAPTER 4

IMPLEMENTATION

4.1 Software and Hardware Requirements

Hardware Requirements

- Processor - Intel 486/Pentium processor or better
- Processor Speed – 1.8 GHz or above
- Hard Disk - 20GB(approx.)
- RAM – 2GB or above
- Storage Space - Approx. 100MB

Software Requirements

- Language Used : C#
- Database : MS SQL Database
- FRONT-End Design : ASP.NET
- Web Browser : Google Chrome, Mozilla
- Software :Visual Studio

C#

C# is a type-safe and sophisticated object-oriented language that allows designers to construct a range of protected and robust applications that work on the .NET Framework. It can be used to develop Windows customer applications, XML Web services, dispersed parts, client-server applications, database applications, etc. In this project we have used C# as back-end software that interacts with the database through Microsoft SQL.

MS SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

ASP.NET

ASP.NET is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

4.2 Discussion of Code Segment

4.2.1 SQL Queries

Login :

```
namespace Charity_Database_Management
{
    public partial class Student_Login : Form
    {
        SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"|DataDirectory|\\Database.mdf\";Integra
ted Security=True");
        SqlCommand cmd;
        SqlDataReader dr;
        private String getUsername()
        {
            //fetch data from database
            con.Open();
            String syntax = "SELECT Email_id FROM Login where Email_id= '"+textBox1.Text
+"'";
            cmd = new SqlCommand(syntax, con);
            dr = cmd.ExecuteReader();
            dr.Read();
            String temp = dr[0].ToString();
            con.Close();
            return temp;
        }
        private String getPassword()
        {
            //fetch data from database
            con.Open();
            String syntax = "SELECT Password FROM Login where Password=
'"+maskedTextBox1.Text+"'";
            cmd = new SqlCommand(syntax, con);
            dr = cmd.ExecuteReader();
            dr.Read();
            String temp = dr[0].ToString();
            con.Close();
            return temp;
        }
        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                String Uname = getUsername(), Upass = getPassword(), name, pass;
                name = textBox1.Text;
                pass = maskedTextBox1.Text;
                if (name.Equals(Uname) && pass.Equals(Upass))
                {
                    //login
                    label3.Hide();
                    MessageBox.Show("Login success");
                    introduction obj = new introduction();
                    this.Hide();
                    obj.Show();
                }
                else
                {
                    //dont login
                    label3.Show();
                }
            }
            catch(Exception es)
            {
                MessageBox.Show("invalid credentials");
            }
        }
    }
}
```

Student Signup :

```
CREATE PROCEDURE [dbo].stusignup1
       @Email_id varchar(30),
       @Password varchar(30)
AS
       insert into Login(Email_id,Password) values(@Email_id,@Password)
RETURN 0

CREATE PROCEDURE stusignup
       @USN varchar(20),
       @Name varchar(20),
       @Branch varchar(10),
       @Semester int,
       @PhoneNo varchar(20),
       @Email_id varchar(30),
       AS
INSERT INTO Student(USN,Name,Branch,Semester,PhoneNo,Email_id)
VALUES(@USN,@Name,@Branch,@Semester,@PhoneNo,@Email_id);

namespace Charity_Database_Management
{
    public partial class Student_signup : Form
    {
        SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"|DataDirectory|\\Database.mdf\";Integra
ted Security=True");
        private void button2_Click(object sender, EventArgs e)
        {
```

```
SqlCommand cmd = new SqlCommand("stusignup1", con);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@Email_id", textBox6.Text);
cmd.Parameters.AddWithValue("@Password", textBox7.Text);
con.Open();
try
{
    cmd.ExecuteNonQuery();
}
catch (Exception ex)
{
    MessageBox.Show("INVALID" + ex);
}
con.Close();
SqlCommand cmd = new SqlCommand("stusignup", con);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@USN", textBox1.Text);
cmd.Parameters.AddWithValue("@Name", textBox2.Text);
cmd.Parameters.AddWithValue("@Branch", textBox3.Text);
cmd.Parameters.AddWithValue("@Semester", textBox4.Text);
cmd.Parameters.AddWithValue("@PhoneNo", textBox5.Text);
cmd.Parameters.AddWithValue("@Email_id", textBox6.Text);
con.Open();
try
{
    cmd.ExecuteNonQuery();
    MessageBox.Show("Sign Up Successful");
}
catch (Exception ex)
{
    MessageBox.Show("INVALID" + ex);
}
con.Close();
```

```
            introduction obj1 = new introduction();
            this.Hide();
            obj1.Show();
        }
    }
}
```

College Staff Signup :

```
CREATE PROCEDURE [dbo].Staffsignup1
       @Email_id varchar(30),
       @Password varchar(30)
AS
       insert into Login(Email_id,Password) values(@Email_id,@Password)
RETURN 0

CREATE PROCEDURE [dbo].Staffsignup
       @EID varchar(20),
       @Name varchar(20),
       @Branch varchar(10),
       @Designation  varchar(20),
       @PhoneNo varchar(20),
       @Email_id varchar(30),
AS
       insert into College_Staff(EID,Name,Branch,Designation,PhoneNo,Email_id)
values(@EID,@Name,@Branch,@Designation,@PhoneNo,@Email_id)
RETURN 0;

namespace Charity_Database_Management
{
    public partial class Staff_Signup : Form
    {
        SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"|DataDirectory|\\Database.mdf\";Integra
ted Security=True");
        private void button1_Click(object sender, EventArgs e)
        {
```

```
SqlCommand cmd = new SqlCommand("Staffsignup1", con);
cmd.Parameters.AddWithValue("@Email_id", textBox6.Text);
cmd.Parameters.AddWithValue("@Password", textBox7.Text);
con.Open();
try
{
    cmd.ExecuteNonQuery();
}
catch (Exception ex)
{
    MessageBox.Show("INVALID" + ex);
}
con.Close();
SqlCommand cmd = new SqlCommand("Staffsignup", con);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@EID", textBox1.Text);
cmd.Parameters.AddWithValue("@Name", textBox2.Text);
cmd.Parameters.AddWithValue("@Branch", textBox3.Text);
cmd.Parameters.AddWithValue("@Designation", textBox4.Text);
cmd.Parameters.AddWithValue("@PhoneNo", textBox5.Text);
cmd.Parameters.AddWithValue("@Email_id", textBox6.Text);
con.Open();
try
{
    cmd.ExecuteNonQuery();
    MessageBox.Show("Sign Up Successful");
}
catch (Exception ex)
{
    MessageBox.Show("INVALID" + ex);
}
con.Close();
introduction obj1 = new introduction();
```

```
            this.Hide();
            obj1.Show();
        }
    }
}
```

Amount Donation:

```
CREATE PROCEDURE [dbo].DonationAdd
       @DNo varchar(15),
       @Email_id varchar(30),
       @CNo varchar(15),
       @Donated_Amt  int,
       @Mode_Of_Payment varchar(30),
       @Card_No varchar(30)

AS
       insert into Donation_Amount(DNo,Email_id,CNo,Donated_Amt,Mode_of_Payment,Card_No)
       values(@DNo,@Email_id,@CNo,@Donated_Amt,@Mode_Of_Payment,@Card_No)
RETURN 0
namespace Charity_Database_Management
{
    public partial class MakeaDonation : UserControl
    {
      private void button1_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = new SqlCommand("DonationAdd", con);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@DNo", textBox2.Text);
            cmd.Parameters.AddWithValue("@Email_id", textBox5.Text);
            cmd.Parameters.AddWithValue("@CNo", textBox4.Text);
            cmd.Parameters.AddWithValue("@Donated_Amt", textBox3.Text);
            cmd.Parameters.AddWithValue("@Mode_of_Payment", textBox11.Text);
            cmd.Parameters.AddWithValue("@Card_No", textBox7.Text);
            con.Open();

            try

            {

                cmd.ExecuteNonQuery();

                MessageBox.Show("Donation Successful");

            }

            catch (Exception ex)

            {

                MessageBox.Show("INVALID" + ex);

            }

            con.Close();

        }
    }
}
```

Stock Donation :

```
CREATE PROCEDURE [dbo].DonationAdd3

@DNum varchar(15),

       @Email_id varchar(30),

       @CNo varchar(15),

       @Donated_Item  varchar(50),

       @No_of_Items int

AS

       insert into Donation_Item(DNum,Email_id,CNo,Donated_Item,No_of_Items)
values(@DNum,@Email_id,@CNo,@Donated_Item,@No_of_Items)

RETURN 0
```

```
namespace Charity_Database_Management

{

    public partial class MakeaDonation : UserControl

    {

      private void button2_Click(object sender, EventArgs e)

        {

            SqlCommand cmd = new SqlCommand("DonationAdd3", con);

            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.AddWithValue("@DNum", textBox12.Text);
```

```
                cmd.Parameters.AddWithValue("@Email_id", textBox6.Text);
                cmd.Parameters.AddWithValue("@CNo", textBox10.Text);
                cmd.Parameters.AddWithValue("@Donated_Item", textBox9.Text);
                cmd.Parameters.AddWithValue("@No_of_Items", textBox8.Text);
                con.Open();
                try
                {
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("Donation Successful");
                }
                catch (Exception ex)
                {
                    MessageBox.Show("INVALID" + ex);
                }
                con.Close();
            }
        }
}
```

4.3Applications of Fund Raising Database Management System

- This Project helps in maintaining clean records of the funds recorded and the various sectors where the funds are distributed. This allows transparency in the system and encourages individuals to contribute without the fear of their money being misused. There must be clear accounts of the funds pooled in the so far, and the goal to be achieved.

- This Project helps to classify and categorize Charities under standard parameters which are applicable globally.

- This Project can also be used to obtain statistics of various attributes so that, there can be improvements made in the particular sectors where the funds are relatively less.

4.4 Discussion of Results

Snapshots

Login page

The figure **4.1** refers to the Login Page. The login category page directs the user to enter whether he or she is a student or a member of the College staff. This page allows the user to either to Login or Signup. The buttons redirect the user to page respectively. This project Allows students as well as the staff to maintain an account.

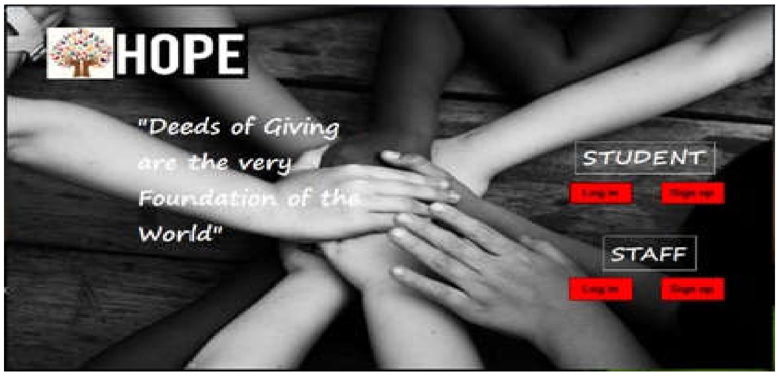


Figure 4.1: Login page

Student Login page

The Figure **4.2** refers to the student login page. This page is designed for the user to enter the credentials and login into the account. The query written for this page is used to compare the email-id and password saved in the database with the values entered in the textbox respectively. If the query fetches the required information from the database and lets the user login depending whether his credentials are correct or not.

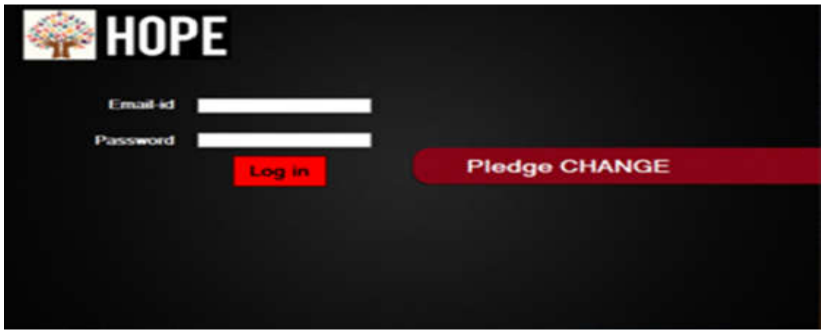


Figure 4.2: Login page

Student Signup page

The Figure **4.3** refers to the Student Signup page. When the user clicks the Signup button under the student section, he/she is redirected to this page. The student has to enter the USN, Name, Branch, Semester, Phone number, Email-ID and a password of one's own choice. This creates an account for the student and allows him/her to explore further pages of the project.

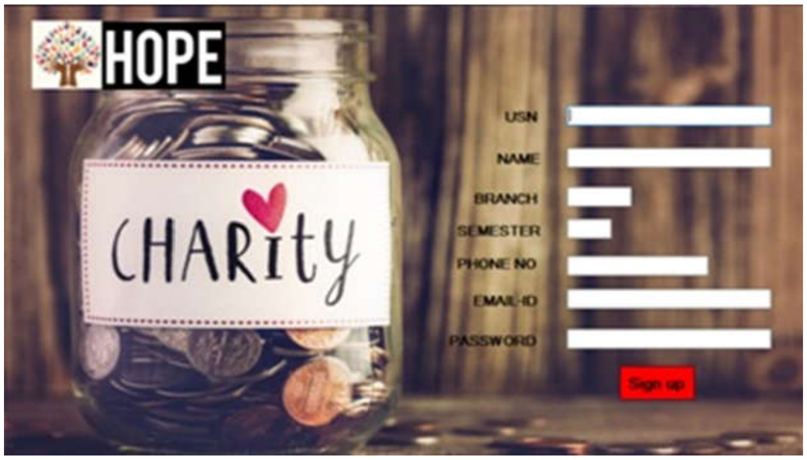


Figure 4.3: Student Signup Page

Clg_Staff Login page

The Figure **4.4** refers to the College Staff login page. This page is designed for the user to enter the credentials and login into the account. The query written for this page is used to compare the email-id and password saved in the database with the values entered in the textbox respectively. If the query fetches the required information from the database and lets the user login depending whether his credentials are correct or not.

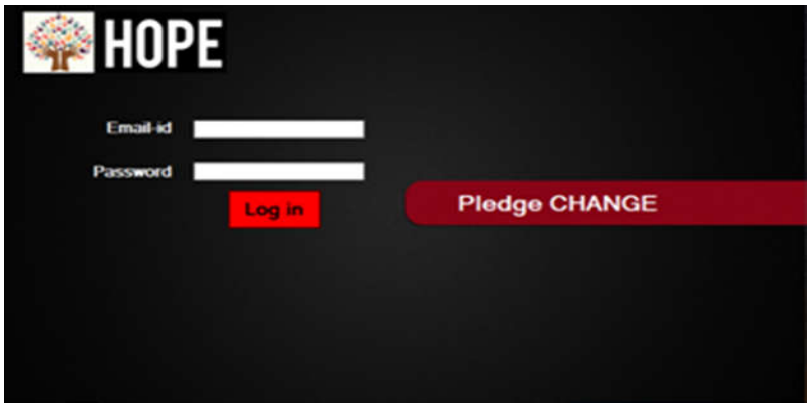


Figure 4.4: College Staff Login Page

Clg_Staff Signup Page

The Figure **4.5** refers to the College Staff Signup page. When the user clicks the Signup button under the student section, he/she is redirected to this page. The student has to enter the EID, Name, Branch, Designation, Phone number, Email-ID and a password of one's own choice. This creates an account for the student and allows him/her to explore further pages of the project.

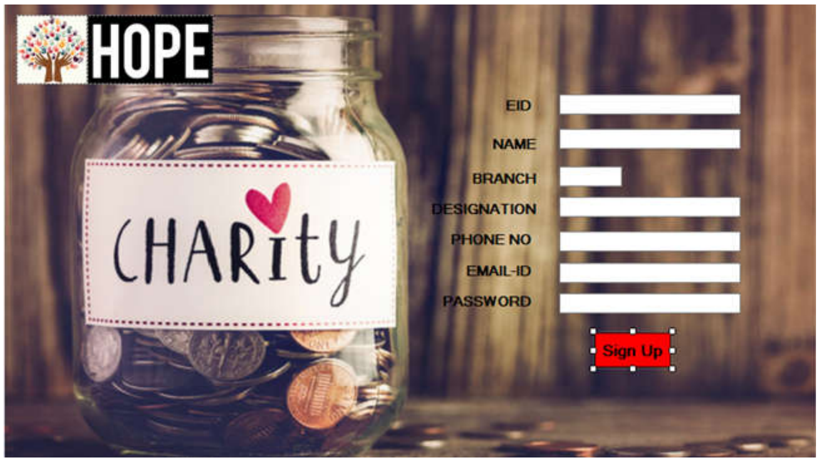


Figure 4.5: College Staff signup Page

Home page

The Figure **4.6** refers to the Home page of the project. The user is redirected to this page as soon as he logs in. There are several buttons on the sliding panel. These buttons are causes we support, donors, about us, help, details and make a donation.

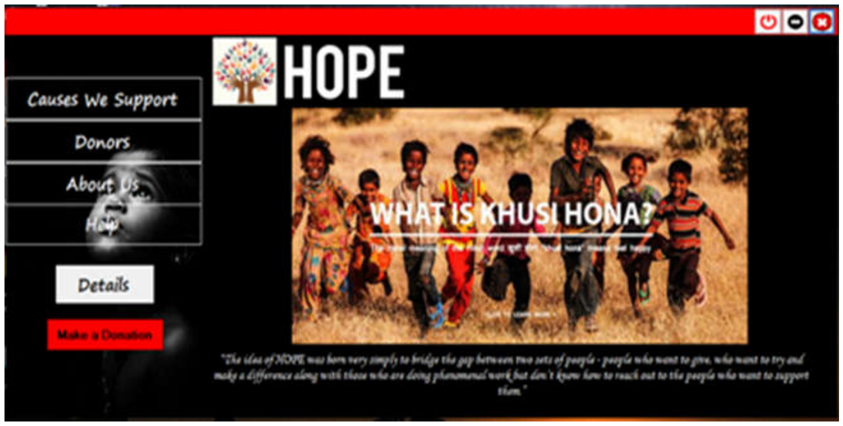
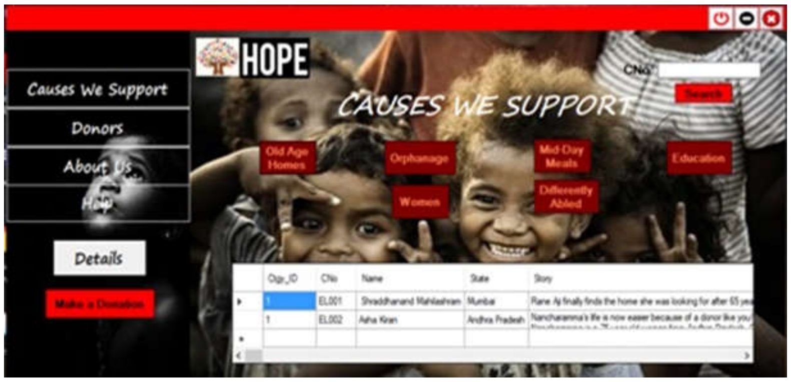


Figure 4.6: Home Page

Charity_Category Page

Figure **4.7** refers to the Charity Category page. Causes we support is the button present on the sliding panel which,on being pressed, displays a set of the causes That this project supports. Namely Old Age homes, Orphanage, Mid-day meals, Education, Women, Differently – abled.On clicking on any one of the causes we support, there will be a list of programs displayed which will let you know the charity Number , story, description, Name, number of Current supporters, raised money, and the Indispensable items for the particular donor Recipient.



Org_ID	CNo	Name	State	Story
1	EL001	Shraddhanand Mahilashram	Mumbai	Rane Aji finally finds the home she was looking for after 65 yea
1	EL002	Asha Kiran	Andhra Pradesh	Nancharamma's life is now easier because of a donor like you!

Figure 4.7: Charity_Category Page

Donor page

Figure **4.8** refers to the Donor page. Under the causes we support is an option called donor. In our project, we can donate either money or stock items such as clothes, food items, electronic gadgets, utensils or any other household items. This page displays two options a list of the donations and a list of the donation stock.

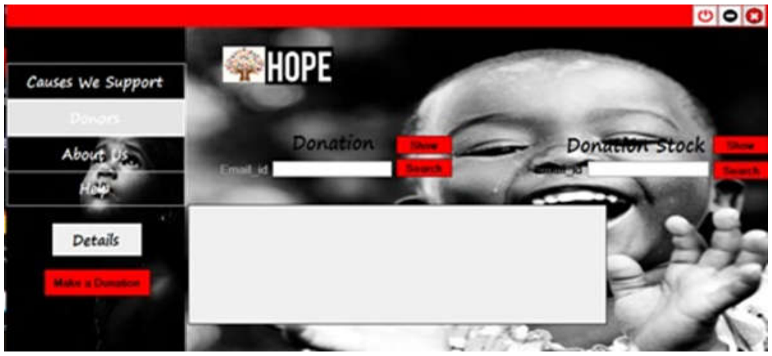


Figure 4.8: Donor Page

Donation page

Figure **4.9** refers to the Donations Page. When the user enters this page, he can make either a donation in the form of money or in the form of a stock item. To donate funds, the user has to enter a unique Donation number, an Email-ID, the Charity number which represents the donor recipient, mode of payment and card number. For donation stock, the user has to enter the donation number, Email-ID, the charity number, name of the item and the number of items.

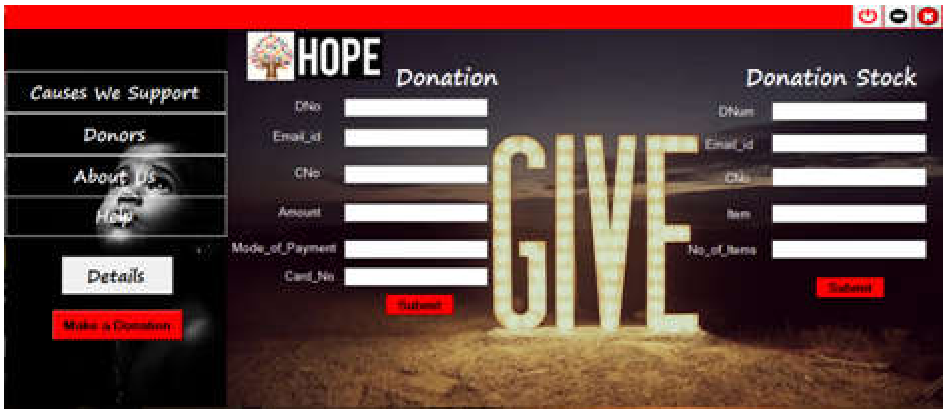


Figure 4.9: Donations Page