**WEB APPLICATION FOR REGISTRY FUNCTIONALITIES**

A PROJECT REPORT

**.NET PROGRAMMING**

*by*

**VAIBHAV KAKKAR (15BIT0199)**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **TITLE** | **PAGE NO.** |
| **1** | **Abstract** | 4 |
| **2** | **Introduction** | 5 |
| **3** | **Registry Sub key Diversity** | 6 |
| **4** | **Registry Key Properties** | 7 |
| **5** | **Data Flow Diagram** | 8 |
| **6** | **Implementation** | 9 |
| **7** | **Project Coding** | 10 |
| **8** | **Feasibilty of Project** | 25 |
| **9** | **Conclusion** | 25 |
| **10** | **Future Scope** | 25 |

1. **ABSTRACT**

**The Windows registry** acts as a central repository of information for the operating system and the applications on a computer. This database is organized in a hierarchical format, based on a logical ordering of the elements stored within it. When storing information in the registry, select the appropriate location based on the type of information being stored.

It is necessary to avoid destroying information created by other applications because this can cause those applications to exhibit unexpected behaviour and can adversely affect your own application. This project proposes a web tool for registry with various functionalities. The various functionalities include creation, deletion, updation or deletion of registry subkeys.

1. **INTRODUCTION**

Registry keys are the base unit of organization in the registry; they can be compared to folders in Windows Explorer. A particular key can have subkeys (just as a folder can have subfolders). Each key can also have multiple values associated with it, which are used to store information about your application. Each value holds one particular piece of information, which can be retrieved and updated when required. For instance, you can create a registry key for your company under the key HKEY\_LOCAL\_MACHINE\Software and then a subkey for each application that your company creates. Each subkey holds information specific to that application such as color settings, screen location, and product-specific file extensions.

Windows NT, 2000, and XP provide two versions of a Registry Editor: Regedt32.exe and Regedit.exe. Regedt32.exe is automatically installed in the %systemroot%\System32 folder. Regedit.exe is automatically installed in the %systemroot% folder. You can modify the registry using either of these Registry Editor utilities. However, if possible, you should use other utilities and tools provided with Windows 2000 to modify your system settings, such as those in the Control Panel. When you modify the registry with Registry Editor, the editor does not check for syntax or other errors. In addition, one modification to the registry may cause a cascade of changes throughout it. The results of an incorrect edit made with Registry Editor are unpredictable and may impair or disable the Windows 2000 operating system. However, by using other tools and utilities, you can ensure that modifications made to the registry are logical and valid, and you can manage any subsequent cascade of changes an edit may cause.

You can use Regedt32.exe in read-only mode (on the Options menu, click Read Only Mode) to safely view the registry and not inadvertently make changes. Switch off read-only mode when you are certain of the changes you wish to make.

1. **REGISTRY SUBKEYS DIVERSITY**

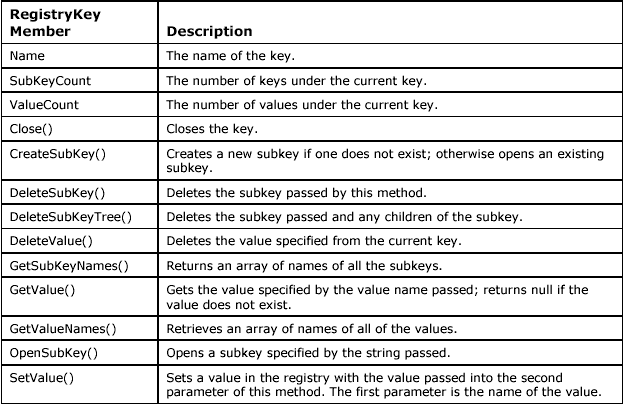
The information stored in the registry is available to other applications and users, and therefore you should not use the registry to store security or critical application information. The main base Registry categories for the Microsoft operating systems are as follows:

* **CurrentUser.** Stores information about user preferences.
* **LocalMachine.** Stores configuration information for the local machine.
* **ClassesRoot.**Stores information about types (and classes) and their properties.
* **Users.**Stores information about the default user configuration.
* **PerformanceData.** Stores performance information for software components.
* **CurrentConfig.**Stores non-user-specific hardware information.
* **DynData.**Stores dynamic data.

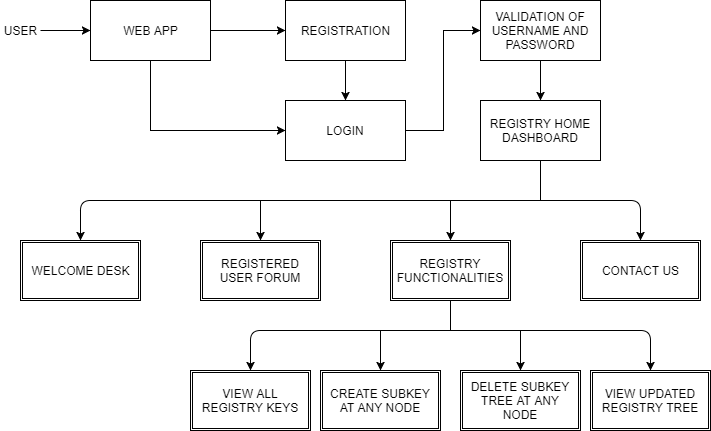
The Registry class has a static field corresponding to each of these key types. The Registry class members are described as follows:

* **ClassesRoot.**Returns a RegistryKey type that provides access to the HKEY\_CLASSES\_ROOT key.
* **CurrentConfig.**Returns a RegistryKey type that provides access to the HKEY\_CURRENT\_CONFIG key.
* **CurrentUser.** Returns a RegistryKey type that provides access to the HKEY\_CURRENT\_USER key.
* **DynData.**Returns a RegistryKey type that provides access to the HKEY\_DYN\_DATA key.
* **LocalMachine.**Returns a RegistryKey type that provides access to the HKEY\_LOCAL\_MACHINE key.
* **PerformanceData.** Returns a RegistryKey type that provides access to the HKEY\_PERFORMANCE\_DATA key.
* **Users.**Returns a RegistryKey type that provides access to the HKEY\_USERS key.

**REGISTRY KEY PROPERTIES**



1. **DATA FLOW DIAGRAM**



1. **IMPLEMENTATION**

**System requirements- server side**

Operational system: Unix/Linux

• Programming language: C#, ASP .NET

• Disk Space on the local server: 100 Mb

**System requirements – client**

• Operation System: Windows/OS/Unix/Linux

• Browser: any browser (Google Chrome, IE 5.5+, Netscape.).

**SOFTWARE AND HARDWARE USED**

This section describes the software and hardware requirements of the system

**SOFTWARE USED**

 **VISUAL STUDIO** : This is Programming software that allows to create various interfaces, web applications, console applications using various functionalities in different various languages.

 **ASP SERVER**

1. **PROJECT CODING – SAMPLE CODES**

**Register**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Microsoft.Win32;

namespace RegistryProject

{

public partial class Register : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Reg\_Click(object sender, EventArgs e)

{

if (RegPass2.Text.Equals(RegPass.Text))

{

RegistryKey rk = Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\ProjectLogin\\login\\User-" + RegUname.Text,true);

if (rk == null)

{

rk = Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\ProjectLogin\\login\\User-" + RegUname.Text);

}

rk.SetValue("Fname", RegName.Text);

rk.SetValue("Email", RegEmail.Text);

rk.SetValue("Uname", RegUname.Text);

rk.SetValue("Pass", RegPass.Text);

Response.Redirect("Login.aspx");

}else

{

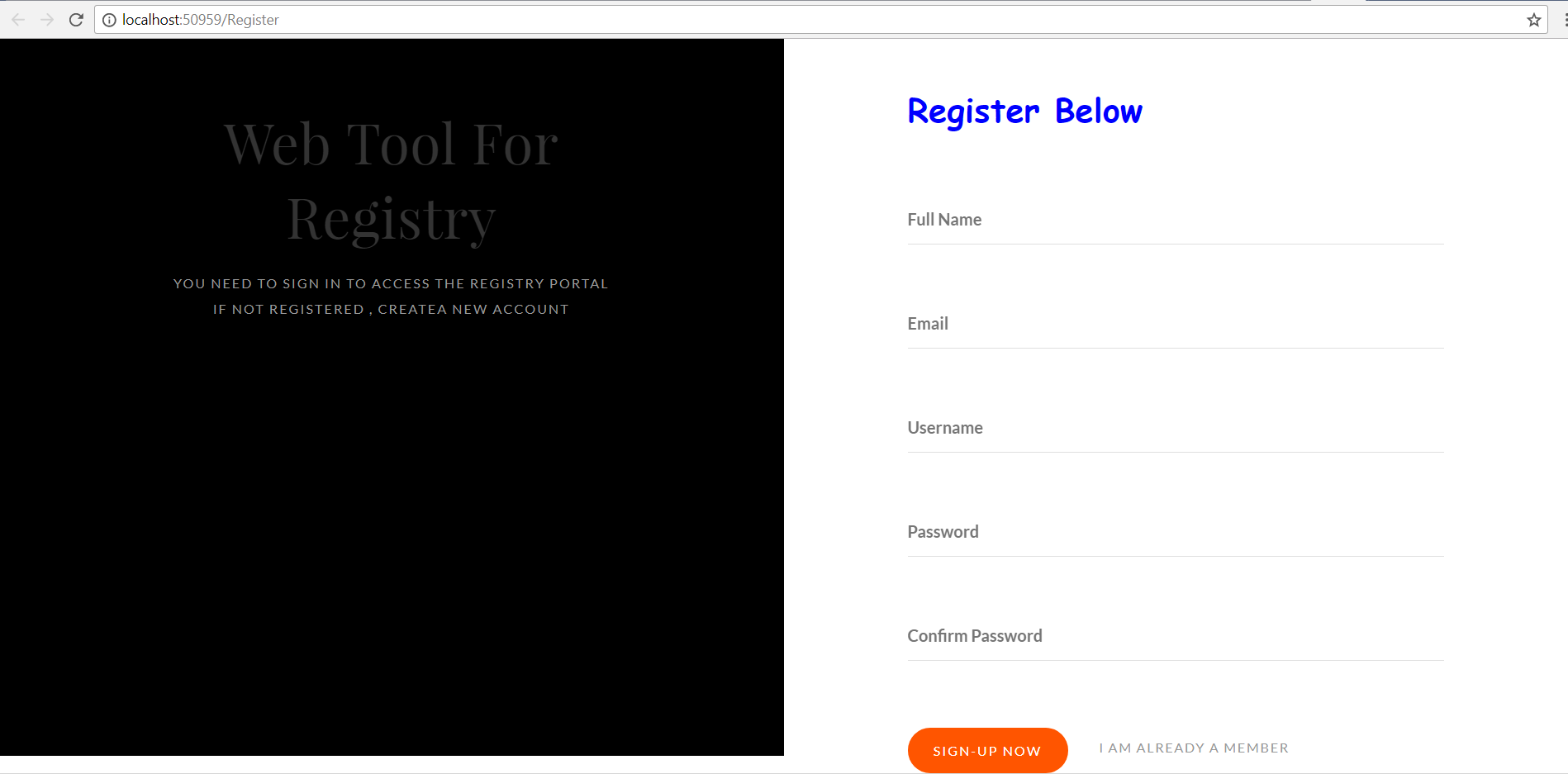
ScriptManager.RegisterClientScriptBlock(this.Page, this.GetType(), "Myscript1", @"alert('Confirm Password don't match !!');", true);

}

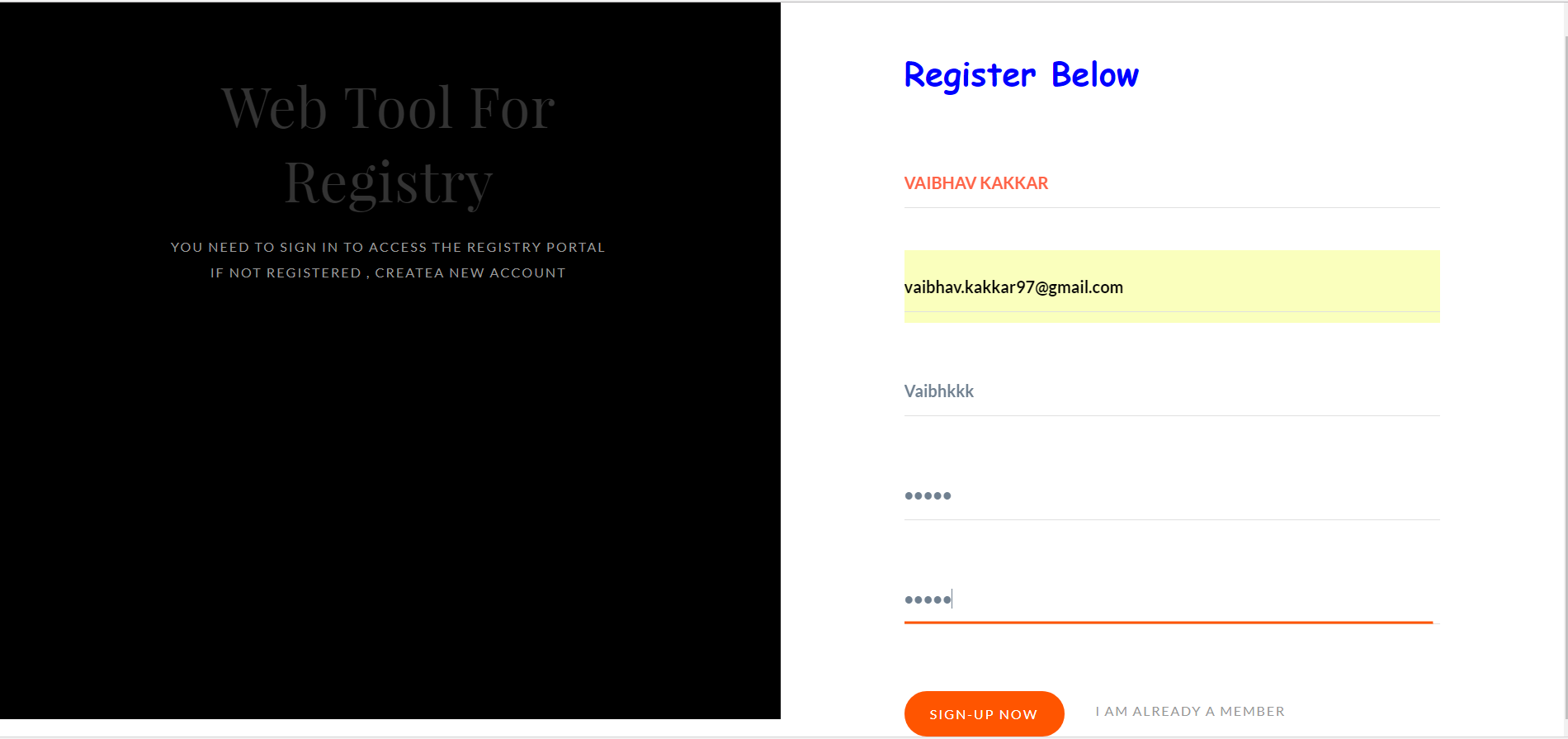
}

}

}



Example: Register



**Login**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Microsoft.Win32;

namespace RegistryProject

{

public partial class Login : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Login\_Click(object sender, EventArgs e)

{

RegistryKey rk = Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\ProjectLogin\\login\\User-" + LogUname.Text, true); ;

if (rk == null)

{

ScriptManager.RegisterClientScriptBlock(this.Page, this.GetType(), "Myscript1", @"alert('UserName not Found , Account doesn't exists ');", true);

}

else

{

if (rk.GetValue("Uname").Equals(LogUname.Text))

{

if (rk.GetValue("Pass").Equals(LogPass.Text))

{

Session["FN"] = rk.GetValue("FName");

Session["LN"] = rk.GetValue("Email");

Response.Redirect("Home.aspx");

}

else

{

ScriptManager.RegisterClientScriptBlock(this.Page, this.GetType(), "Myscript1", @"alert('Wrong Password , Try Again !!');", true);

}

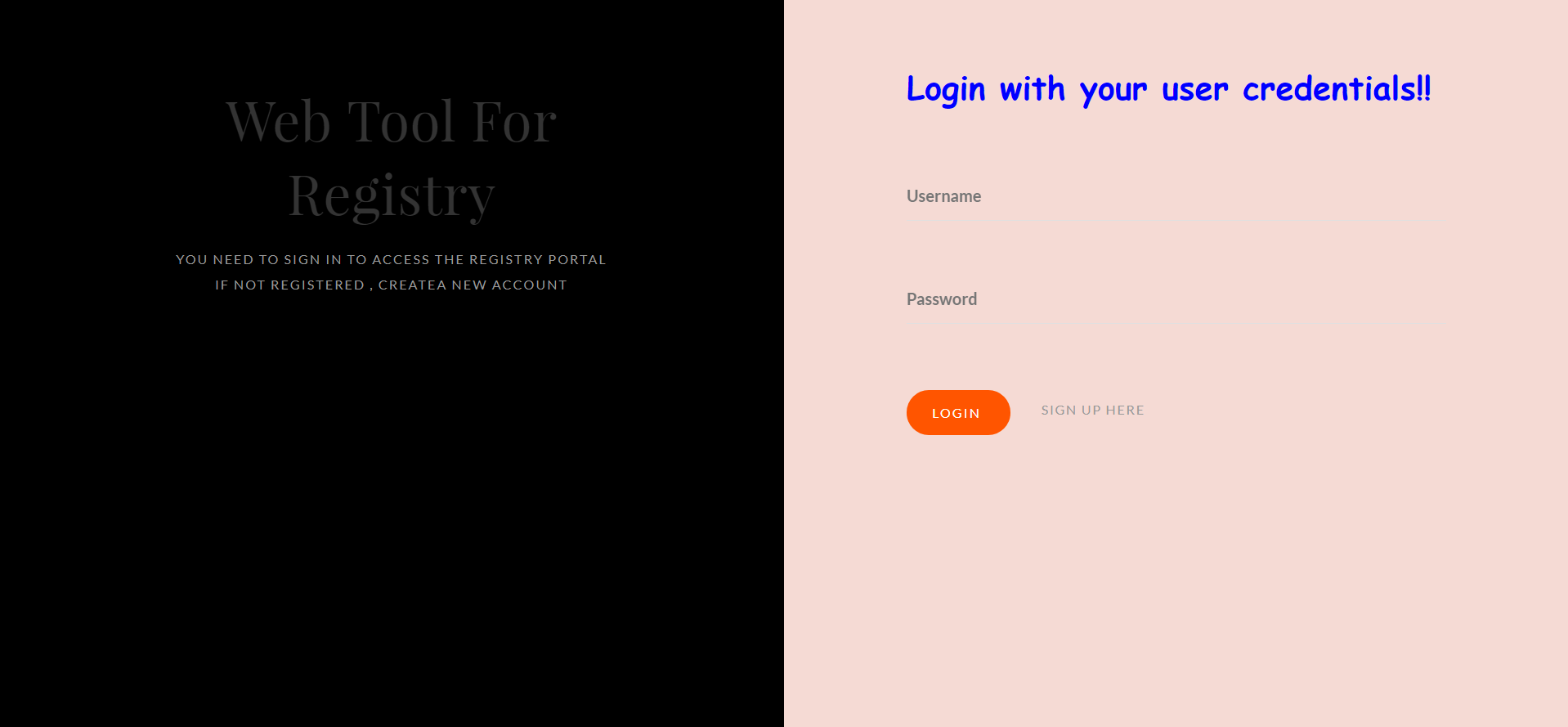
}

}

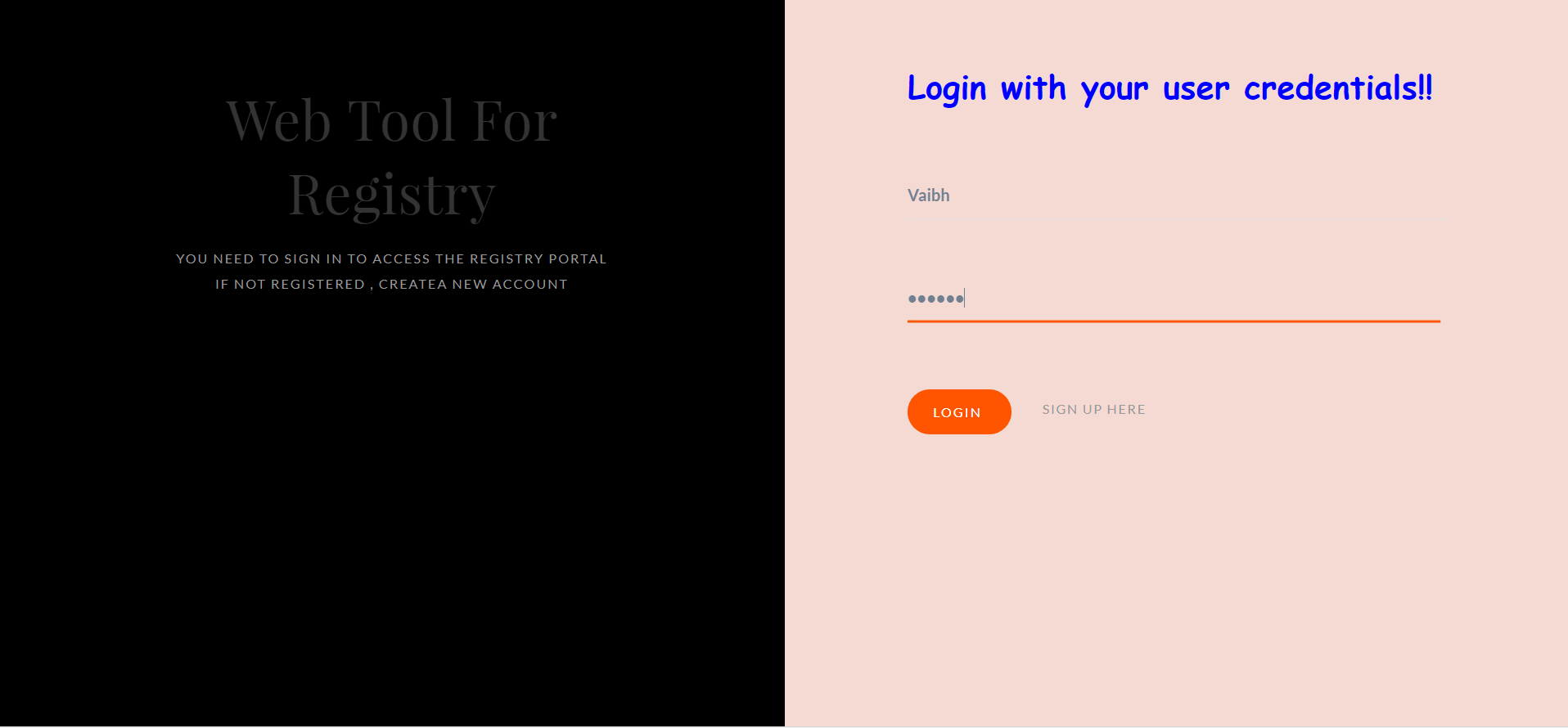
}

}

}



Example: Login



**Home**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Microsoft.Win32;

namespace RegistryProject

{

public partial class Home : System.Web.UI.Page

{

static String pathStr = "";

static String AddToPath = "";

protected void Page\_Load(object sender, EventArgs e)

{

String s = (String)Session["FN"];

HomeName.Text = Session["FN"].ToString() +" "+ Session["LN"].ToString();

RegistryKey rk = Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\ProjectLogin\\login");

string x = "";

string[] keys = rk.GetSubKeyNames();

for (int i = 0; i < rk.SubKeyCount; i++)

{

String keyName = keys[i];

x = x + "<p > <div style='color:red;text-transform: uppercase; font-style: italic;font-weight: bold;'>" + keyName +"</div> ";

RegistryKey rka = Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\ProjectLogin\\login\\"+keyName);

string[] subkeys = rka.GetValueNames();

for (int j = 0; j < subkeys.Length; j++)

{

String subkeyName = subkeys[j];

x = x+" " + subkeyName+"-"+rka.GetValue(subkeyName)+";"; }

x = x + " </p>";

}

AAAA.Text = x;

}

protected void Show\_current()

{

}

protected void Path\_Load(object sender, EventArgs e)

{

}

protected void GOTO\_Click(object sender, EventArgs e)

{

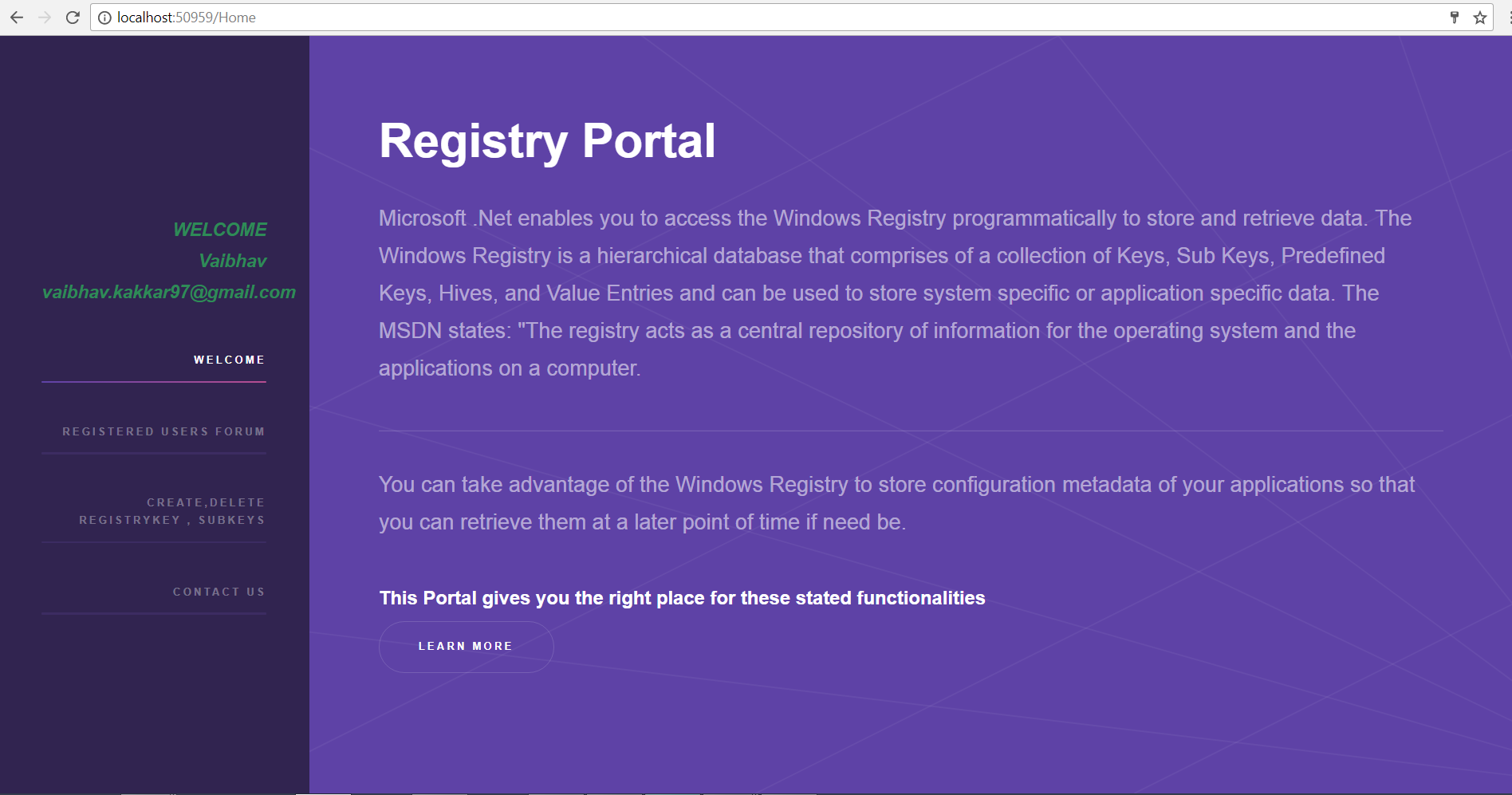
Response.Redirect("RegistryTree.aspx");

}

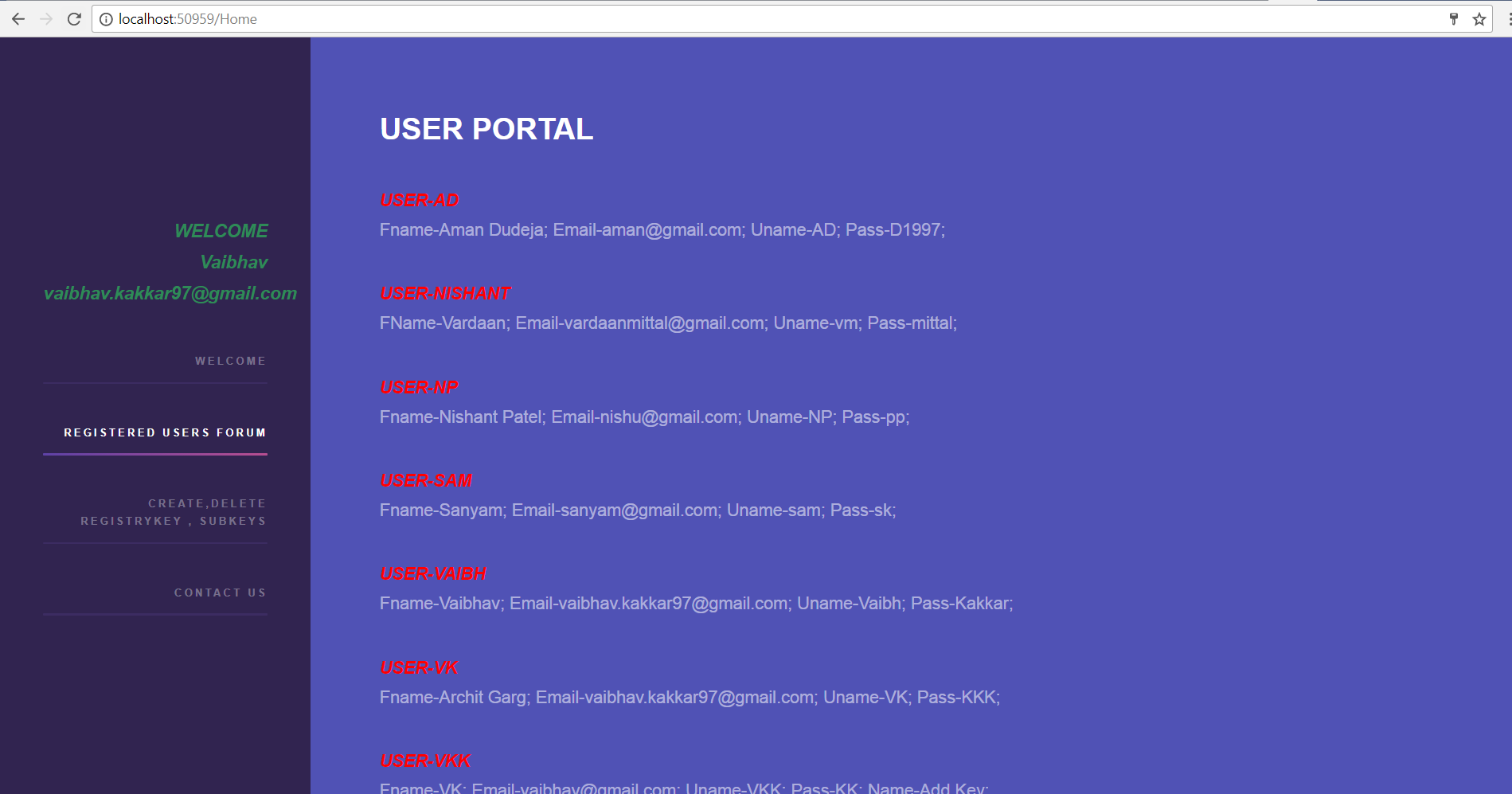
}

}

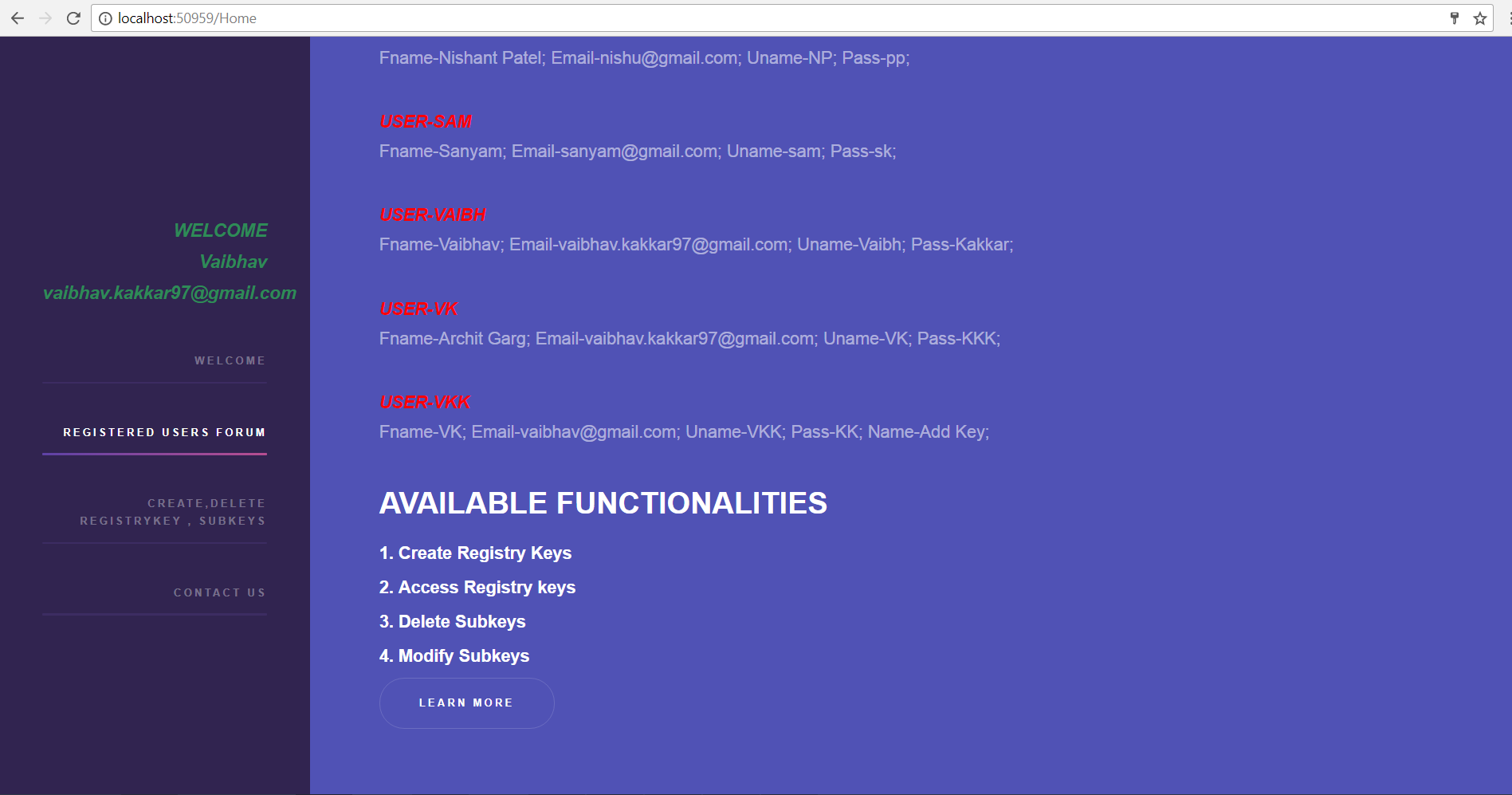
1. WELCOME REGISTRY PORTAL



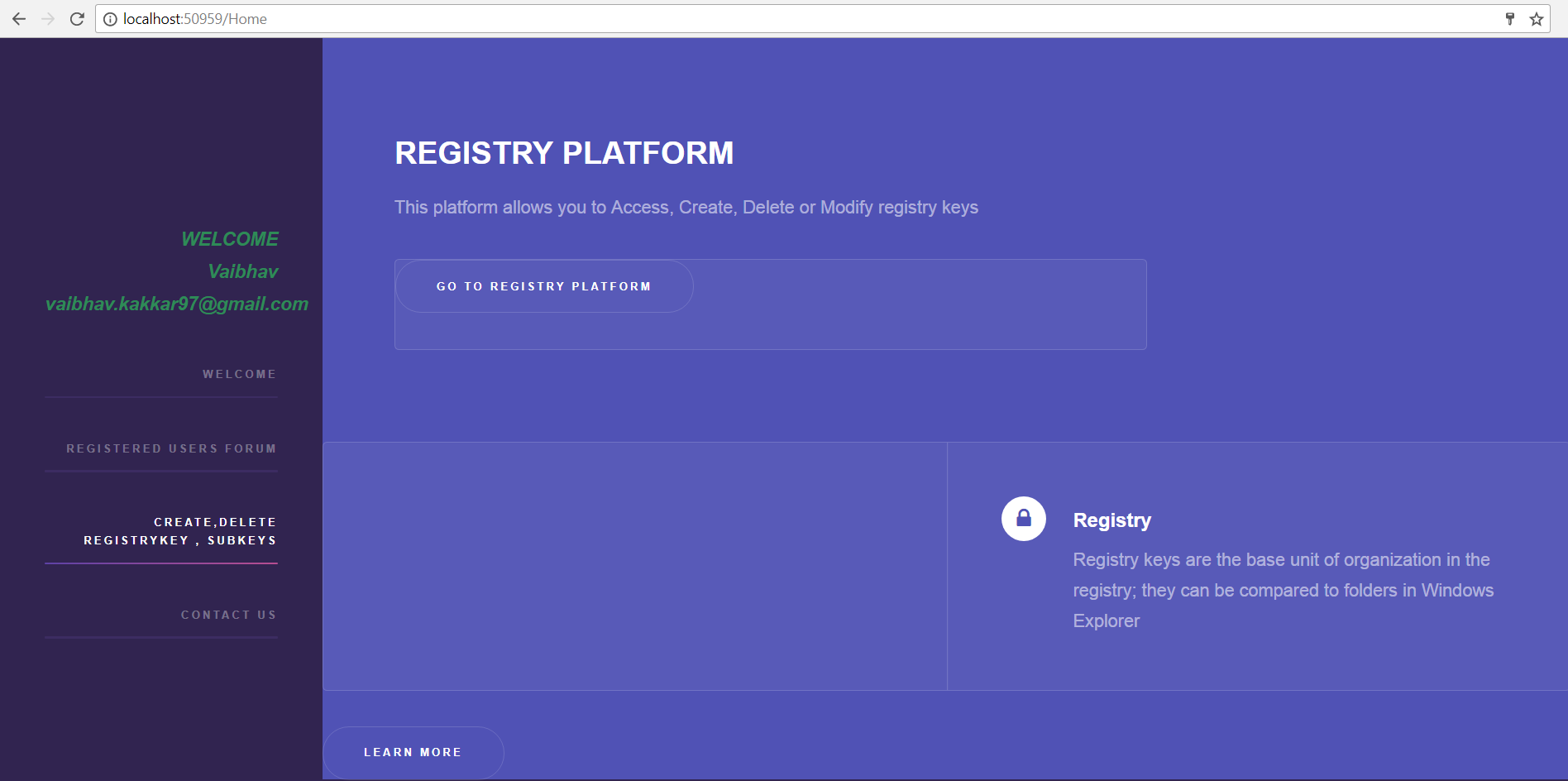
1. REGISTERED USER FORUM



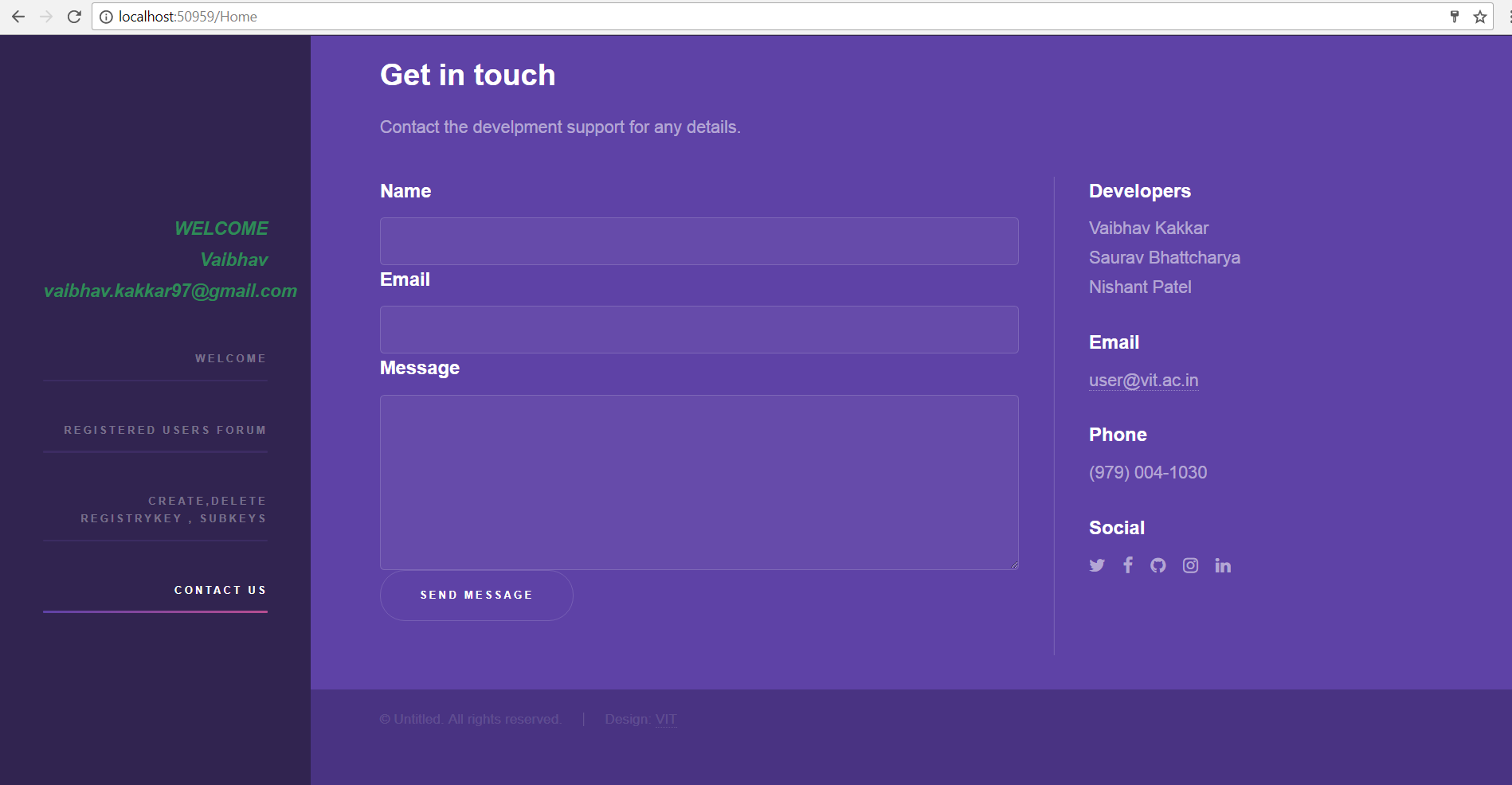
1. AVAILABLE FUNCTIONALITIES



1. REGISTRY FUNCTIONALITY PLATFORM



1. CONTACT US



**REGISTRY PLATFORM**

**REGISTRY TREE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Microsoft.Win32;

namespace RegistryProject

{

public partial class RegistryTree : System.Web.UI.Page

{

static String pathStr = "";

static String AddToPath = "";

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void TreeView1\_SelectedNodeChanged(object sender, EventArgs e)

{

getPath();

}

protected string getPath()

{

TreeNode node = this.TreeView1.SelectedNode;

pathStr = node.Text;

string separator = "\\";

TreeView1.PathSeparator = Convert.ToChar(separator);

while (node.Parent != null)

{

pathStr = node.Parent.Text + separator + pathStr;

node = node.Parent;

}

Path.Text = pathStr;

return pathStr;

}

protected void ShowAll\_Click(object sender, EventArgs e)

{

Update();

}

protected void Update()

{

if (TreeView1 != null)

{

TreeNode tn = new TreeNode("ProjectLogin");

TreeView1.Nodes.Add(tn);

Show("\\ProjectLogin", tn);

}

}

protected void ExpandAll\_Click(object sender, EventArgs e)

{

TreeView1.ExpandAll();

}

protected static void Show(String path, TreeNode x)

{

RegistryKey rkz = Registry.CurrentUser.OpenSubKey("Software\\Microsoft" + path);

try {

string[] subkeys = rkz.GetSubKeyNames();

if (subkeys.Length == 0)

{

string[] subNames = rkz.GetValueNames();

for (int j = 0; j < subNames.Length; j++)

{

String str = subNames[j];

TreeNode n = new TreeNode(subNames[j]);

x.ChildNodes.Add(n);

String t = (String)rkz.GetValue(str);

n.ChildNodes.Add(new TreeNode(t));

}

return;

}

else

{

for (int i = 0; i < subkeys.Length; i++)

{

TreeNode a = new TreeNode(subkeys[i]);

x.ChildNodes.Add(a);

String pp2 = path;

Show(path += "\\" + subkeys[i], a);

path = pp2;

}

}

return;

}

catch(Exception e)

{

e.GetType();

}

}

protected void Show\_current()

{

}

protected void Path\_Load(object sender, EventArgs e)

{

}

protected void AddKey\_Click(object sender, EventArgs e)

{

TreeView1.Nodes.Clear();

Update();

}

protected void Btn\_Delete\_Click(object sender, EventArgs e)

{

RegistryKey add = Registry.CurrentUser.OpenSubKey("Software\\Microsoft" + pathStr,true);

String[] s = pathStr.Split('\\');

String k = "";

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

{

k = k + "\\" + s[i];

}

Path.Text = k;

RegistryKey add2 = Registry.CurrentUser.OpenSubKey("Software\\Microsoft" + k,true);

try

{

add2.DeleteSubKeyTree(s[s.Length - 1]);

}

catch

{

add2.DeleteValue(s[s.Length - 1]);

}

TreeView1.Nodes.Clear();

Update();

}

protected void Button1\_Click(object sender, EventArgs e)

{

if (TB\_Create.Equals(null))

{

ScriptManager.RegisterClientScriptBlock(this.Page, this.GetType(), "Myscript1", @"alert('Text is Empty, Enter the name of subkey to be created !!');", true);

}

else

{

Path.Text = "Software\\Microsoft\\" + pathStr;

RegistryKey add = Registry.CurrentUser.OpenSubKey("Software\\Microsoft" + pathStr);

if (add == null)

{

add = Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\" + pathStr + "\\"+TB\_Create.Text);

Path.Text = "Created";

}

if(T1.Text!="" & T2.Text != "")

{

add.SetValue(T1.Text, T2.Text);

}

}

TreeView1.Nodes.Clear();

Update();

}

protected void Button2\_Click(object sender, EventArgs e)

{

RegistryKey add = Registry.CurrentUser.OpenSubKey("Software\\Microsoft" + pathStr, true);

String[] s = pathStr.Split('\\');

String k = "";

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

{

k = k + "\\" + s[i];

}

Path.Text = k;

RegistryKey add2 = Registry.CurrentUser.OpenSubKey("Software\\Microsoft" + k, true);

try

{

add2.DeleteValue(s[s.Length - 1]);

add2.SetValue(s[s.Length - 1], TB\_Update\_Value.Text);

}

catch

{

}

Path.Text = "Software\\Microsoft\\" + pathStr;

TreeView1.Nodes.Clear();

Update();

}

protected void Button1\_Click1(object sender, EventArgs e)

{

RegistryKey add4 = Registry.CurrentUser.OpenSubKey("Software\\Microsoft" + getPath(), true);

add4.SetValue(T1.Text, T2.Text);

}

protected void GoToHome\_Click(object sender, EventArgs e)

{

Response.Redirect("Home.aspx");

}

protected void ExpandAl\_Click(object sender, EventArgs e)

{

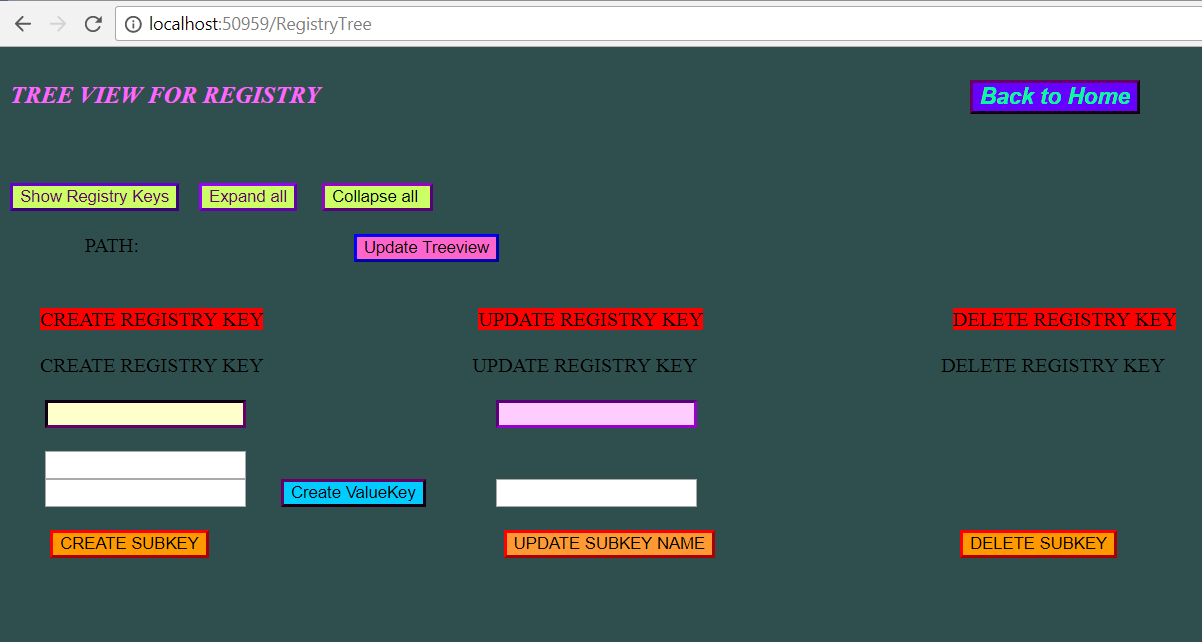
TreeView1.CollapseAll();

}

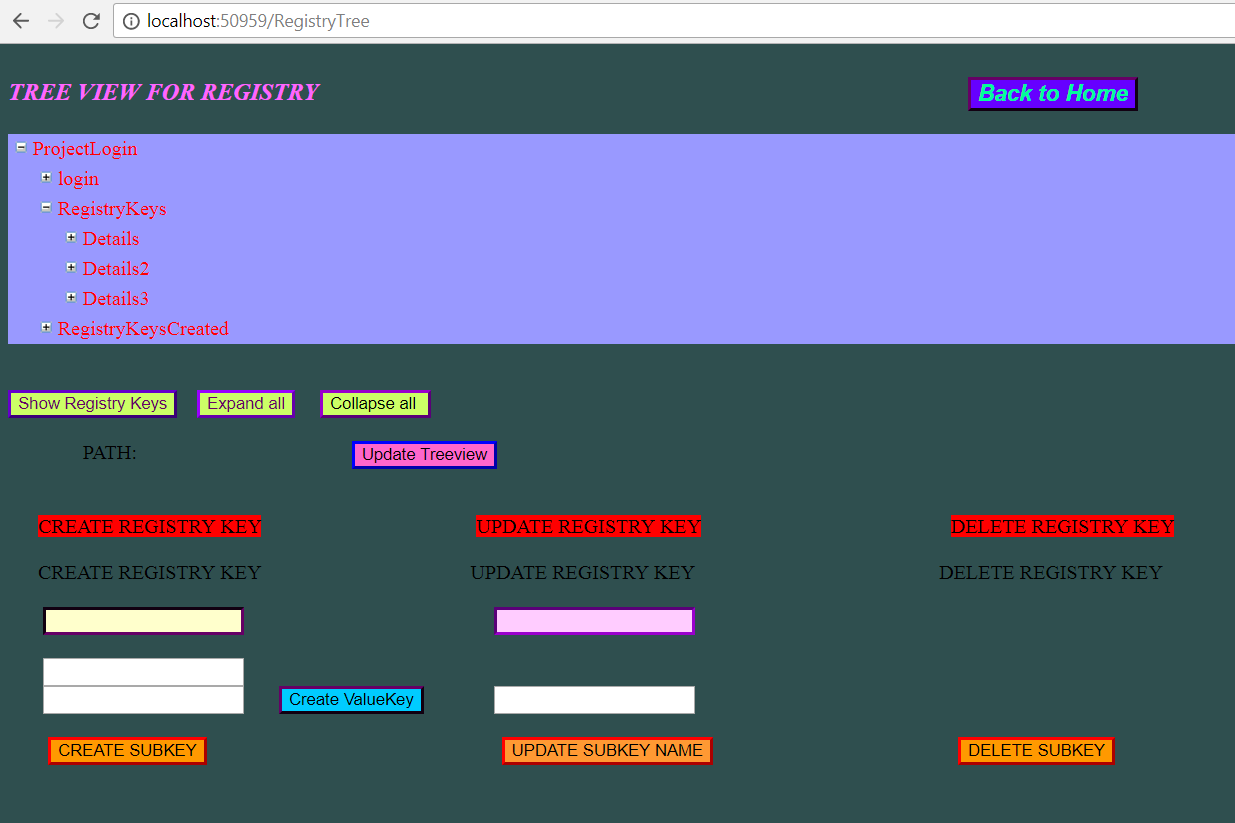
}

}

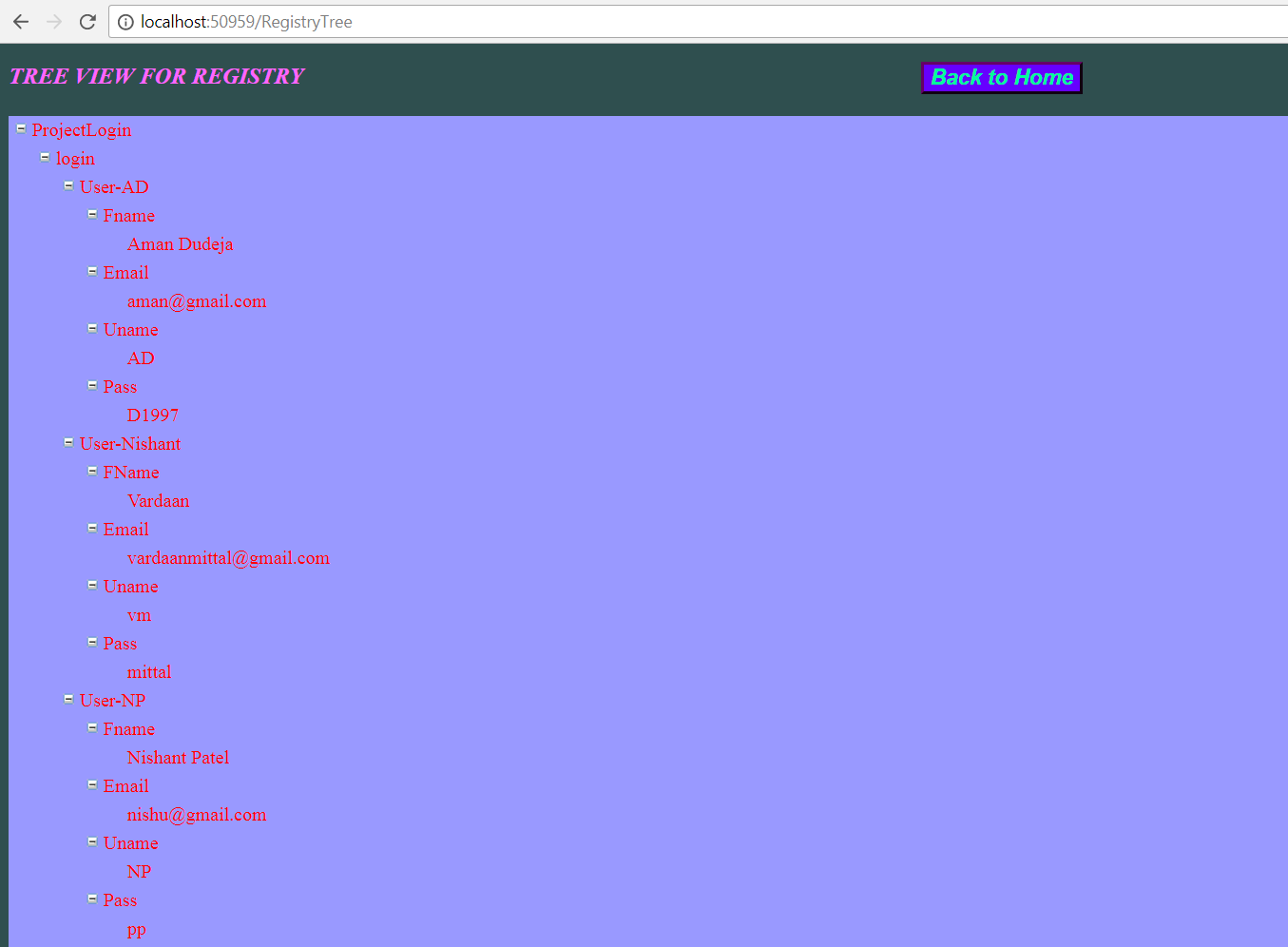
TREEVIEW STRUCTURE



SHOW REGISTRY KEYS

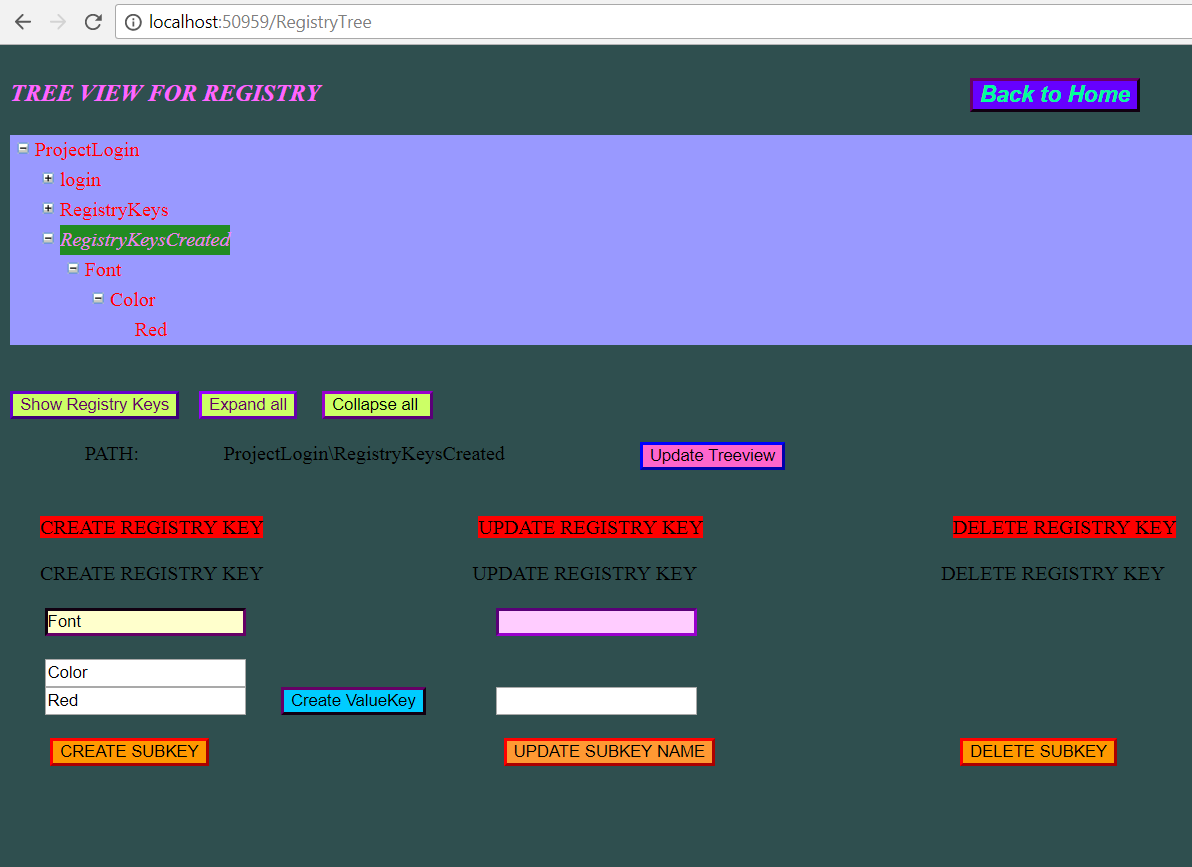


EXPANDED TREEVIEW STRUCTURE

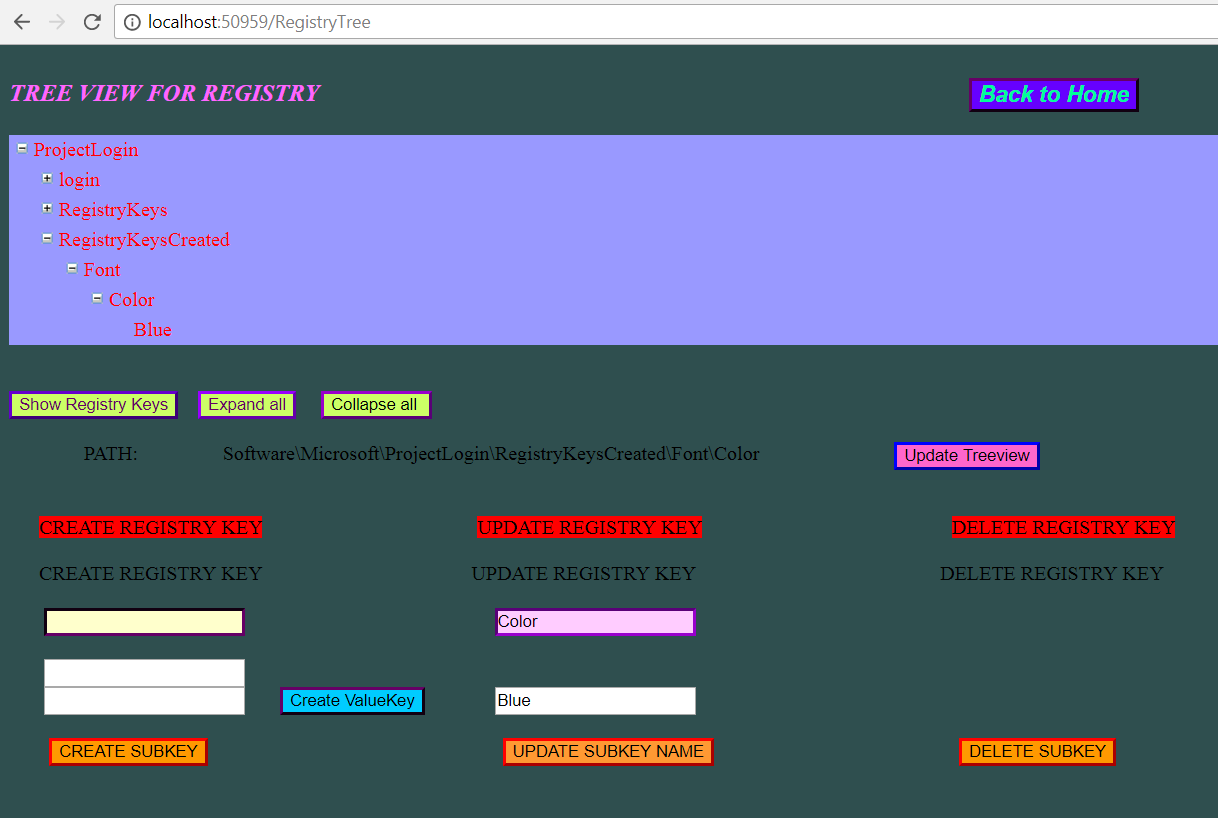


SHOW SELECTED TREENODE PATH

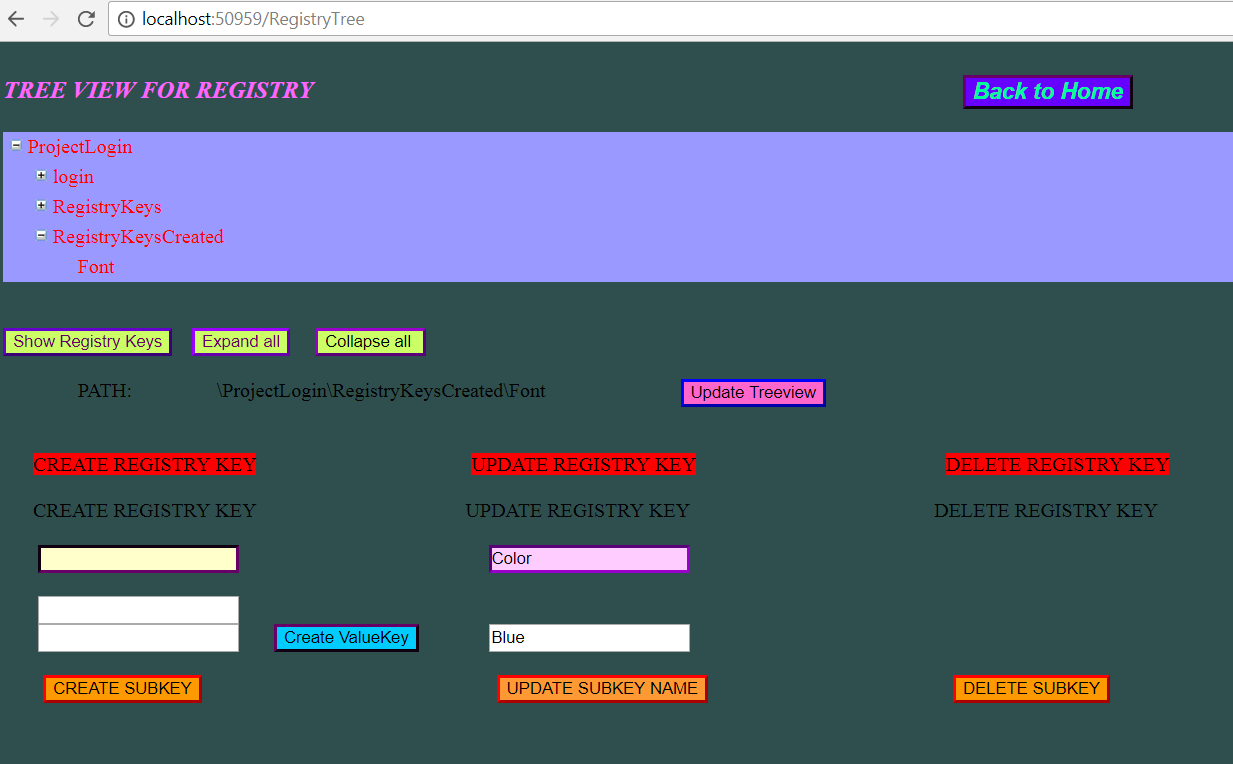
CREATE SUBKEY



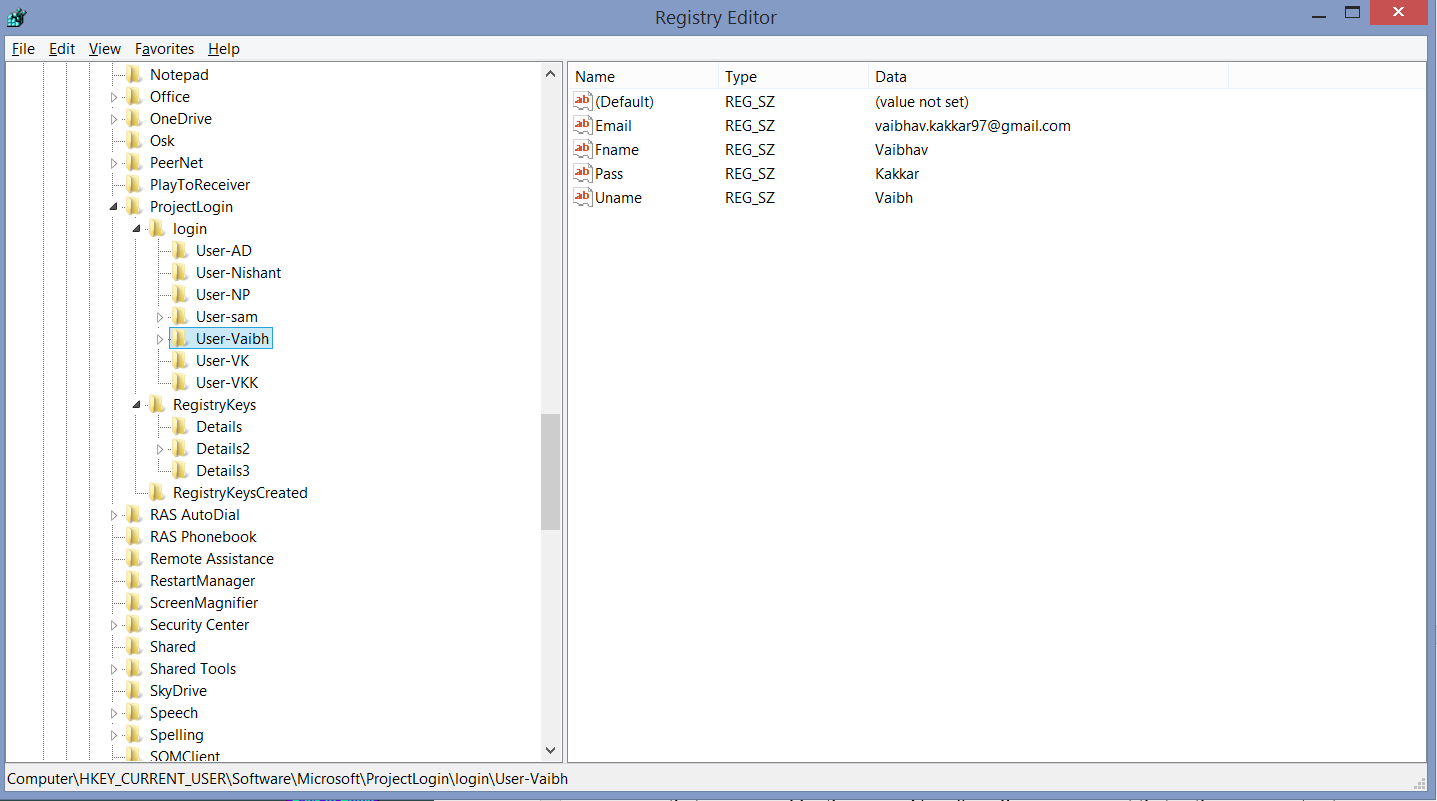
UPDATED TREEVIEW STRUCTURE



DELETE TREENODE



REGEDIT



1. **FEASIBILITY OF PROJECT**

• Technical Feasibility

• Social and Operational Feasibility

• Economic Feasibility

• Time Feasibility

1. **CONCLUSION**

The main aim of developing Registry system is to provide all information that is required by the users about the various details stored about their computers. User friendliness is a must that is the user must get the details without complicated searching procedures. Other important requirements of software are data security, extensibility and maintainability. To maintain these features, we have implemented Registration and Login features as well.

All these features are included in this web application. The project greatly helped in understanding the various phases in web application development and exposure to a new developer platform MS Visual Studio .Net..

We have successfully formed an Web Tool For Registry system with all the necessary features like create, delete , update subkeys. It can perform all the functions of normal class. We have tested the website and can conclude that is fully functional and is very helpful for students to learn the concepts digitally.

1. **FUTURE SCOPE**

• As mentioned above, various enhancement can be made in future.

• We think that our system still has potential to grow. Besides, we will include more functions and introduce more widgets to the system.

We also plan to enhance the interface so that it looks more attractive and interactive