**BCA SEM 6 , YEAR – 2023/2024**

**ONLINE CONSULTANCY PROVIDER**

**Developed By:-**

Rahul Kanara

003203213986

Hit Kanara

003203213985

**Guide By**

Prof. J. M. Ardeshna

Prof. V. M. Isotiya

**College**

M. B. Arts & Commerce College - Gondal

**Submitted To**

Saurashtra University-Rajkot

**Acknowledgement**

The feeling of gratefulness to any one’s help directly arises from the bottom of heart. A small but an important and timely help can prove to be a milestone in one’s life.

Even human being has such kind of experience. Being human being, we also have, the same feeling of gratefulness of today we have achieved an important milestone in my life.

This project is dedicated to all the people, whom we met, took guidance, talked, interviewed and learned something from them. At this occasion we sincerely thank all of them while submitting this project report.

We would like to heartily thank Mr. J. M. Ardeshna & Mr. V. M. Isotiya who has provided us necessary information and guidance for the success of this project.

Last, we would like to extend thanks to all our classmates also. On this occasion we would like to thank Mr. V. M. Isotiya (HOD, Computer Department) without whom this would have not been possible. We are thankful to him for giving time-to-time and valuable guidance during the project period and giving us guidance in taking our curriculum decision and choosing, initiating and getting on with the project.

We always have felt the invisible help from the almighty, without the blessing almighty, we could not have succeeded.

**Index Content**

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Content** | |
| 1 | Project Profile | |
| 2 | System Analysis | |
| 3 | Hardware & Software Specifications | |
| 4 | System Design | |
|  | 4.1 | Data Flow Diagram |
|  | 4.2 | Entity Relationship Diagram |
|  | 4.3 | Data Dictionary |
| 5 | Input and Output Design | |
|  | 5.1 | Screen Shots |
| 6 | Testing | |
| 7 | Future scope and limitations | |
| 8 | Bibliography | |
| 9 | Special Thanks | |

**Project Profile**

|  |  |
| --- | --- |
| **Project Title** | goService : your service provider |
| **Project definition** | A online service provider platform that can users and service providers connects and users take services as their need. |
| **Developed for** | Provided Services Online |
| **Internal project guide** | Mr. Vinod M. Isotiya  Mr. Jaimin S. Ardeshna |
| **Team Size** | 2 |
| **Team Member** | Kanara Rahul L.  Kanara Hit R. |

**Hardware and Software**

Hardware :-( Recommended)

Processor : Intel I3

Random Access Memory : 4 Gb

Hard Disk : 500 Gb

Software :-( Recommended)

Front-End : Visual Studio 2008

Web Server : IIS 5.1

Back End : SQL server 2005

Support : Framework 2.0

Operating System : Windows XP (service pack 2)

Supporting tools : Microsoft Word for document

**Data Flow Diagram**

**System Design**

**ADMIN**

**SERVICE**

**PROVIDER**

**USER**

**User Side Diagram**

**View Services**

**Give**

**Feedback**

**View Ratings**

**View**

**Provider Profile**

**Give ratings**

Can login

As service Provider

**Admin Side Diagram**

View

Rating

Approval

User

Approval

Services

View/Delete

Feedback

**Service Provider Side Diagram**

**ER Diagram**

Edit & Delete

Services

View

Ratings

Add

Services

View & Edit

Profile

|  |
| --- |
| **USER** |
| User id |
| Username |
| Email |
| Password |
| User role |
| ShopName |
| Address |
| Contact Number |
| Image |
| Approval |
| Gender |

|  |
| --- |
| **RATING** |
| Rating id |
| User Name |
| Contact Number |
| Service id |
| Rating Value |
| Additional Info |
| Provider Id |

|  |
| --- |
| **Feedback** |
| Feedback id |
| User Name |
| ContactNumber |
| FeedbackText |
| Data Submit |

|  |
| --- |
| **SERVICE** |
| Service Id |
| Service Name |
| Description |
| Category |
| Subcategory |
| Location |
| Price |
| Provider Id |
| Approval Status |
| Image Path |

**Data Dictionary**

|  |  |
| --- | --- |
| **Table Name** | **Table Known for** |
| User | Registration(Login) |
| Feedback | Give Feedbacks |
| Services | Add Services |
| Rating | Show / Give Rating |

**Data Tables**

* **Table Name:- User**

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Data Type | Length | Constraints |
| UserID | int | 30 | Primary Key |
| UserName | Varchar | 15 | Not null |
| Email | varchar | 20 | Not null |
| Password | Varchar | 20 | Not null |
| UserRole | Varchar | 20 | Not null |
| ShopName | Varchar | 20 | Not null |
| Address | Varchar | 15 | Not null |
| ContactNumber | Varchar | 6 | Not null |
| Image | Varchar | 100 | Not null |
| ApprovalStatus | Varchar | 100 | Not null |
| Gender | Varchar | 1 | Not null |

* **Table Name:-Feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| FieldName | Data Type | Length | Constraints |
| FeedbackID | Int | - | Primary Key |
| UserName | Varchar | 15 | Not null |
| ContactNumber | Varchar | 13 | Not null |
| FeedbackText | text | - | Not null |
| DateSubmitted | datetime | - | Not null |

* **Table Name:-Services**

|  |  |  |  |
| --- | --- | --- | --- |
| FieldName | Data Type | Length | Constraints |
| ServiceID | int | - | Primary Key |
| ServiceName | Varchar | 50 | Not null |
| Description | text | - | Not null |
| Category | Varchar | 20 | Not null |
| Subcategory | Varchar | 20 | Not null |
| Location | Varchar | 20 | Not null |
| Price | Decimal | 10,2 | Not null |
| ProviderID | int | - | Foregin Key |
| ApprovalStatus | Varchar | 10 | Not null |
| ImagePath | Varchar | 100 | Not null |

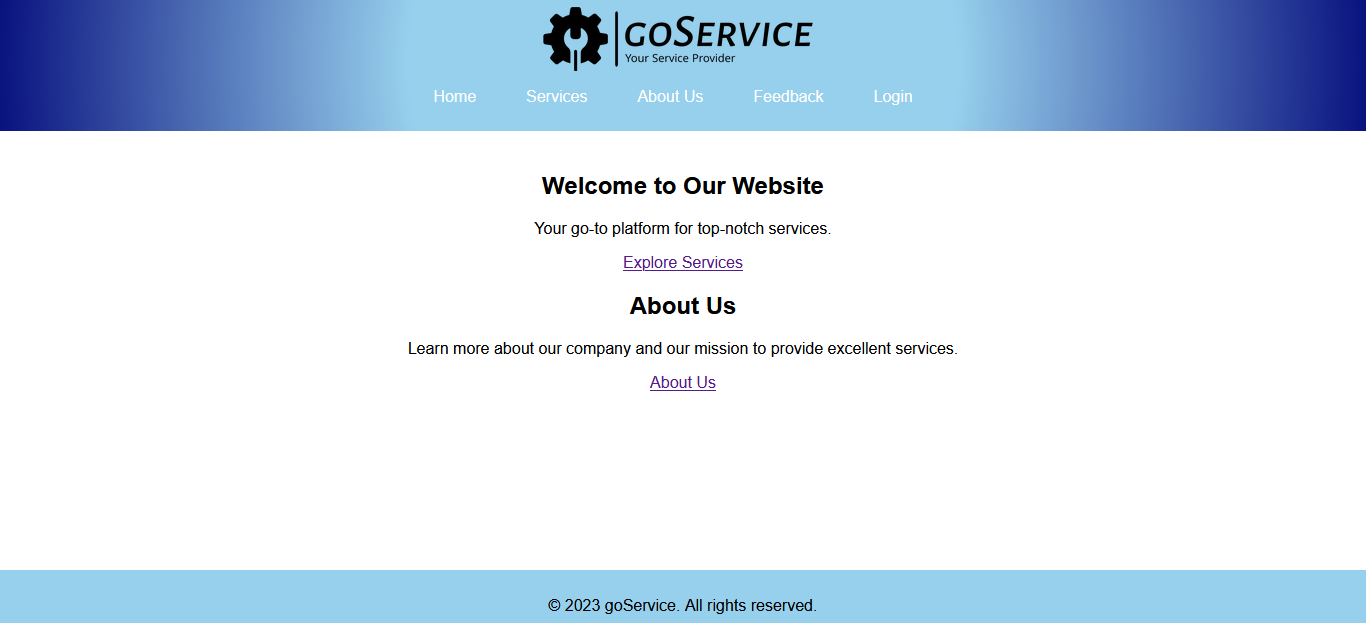
* **Table Name:-Rating**

|  |  |  |  |
| --- | --- | --- | --- |
| FieldName | Data Type | Length | Constraints |
| RatingID | int | - | Primary Key |
| UserName | Varchar | 50 | Not null |
| ContactNumber | Varchar | 13 | Not null |
| ServiceID | int | - | Foregin Key |
| RatingValue | int | - | Not null |
| AdditionalInfo | varchar | 255 | Not null |
| ProviderID | int | - | Foregin Key |

**-: Input & Output Design :-**

**USER SIDE**

**Home Page**

****

**:=>** **Default.aspx**

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div class="main-content">

<section class="welcome-section">

<div class="container">

<h1>Welcome to Our Website</h1>

<p>Your go-to platform for top-notch services.</p>

<a href="Services.aspx" class="cta-button">Explore Services</a>

</div>

</section>

<section class="about-us-section">

<div class="container">

<h2>About Us</h2>

<p>Learn more about our company and our mission to provide excellent services.</p>

<a href="AboutUs.aspx" class="cta-button">About Us</a>

</div>

</section>

</div>

</asp:Content>

**:=>** **Default.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

public partial class \_Default : System.Web.UI.Page

{

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

{

}

}

**:=>** **MasterPage.master**

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head id="Head1" runat="server">

<title>goService</title>

<style>

body {

margin: 0;

padding: 0;

font-family: 'Arial', sans-serif;

}

#master-container {

display: flex;

flex-direction: column;

height: 100vh;

}

header {

background: rgb(151,208,237);

background: radial-gradient(circle, rgba(151,208,237,1) 40%, rgba(9,18,127,1) 100%);

padding: 10px;

text-align: center;

}

.website-name

{

font-family: cursive;

display: flex;

align-items: center;

justify-content: center;

}

.website-name img {

width: 20%;

margin-right: 10px;

}

nav ul {

list-style: none;

display: flex;

justify-content: center;

padding: 0;

}

nav ul li {

margin-right: 20px;

}

nav a {

text-decoration: none;

color: #fff;

padding: 8px 15px;

transition: background-color 0.3s;

border-radius: 5px;

}

nav a:hover {

background-color: #09127f;

}

.content {

text-align: center;

margin: 20px;

flex: 1;

overflow-y: auto;

}

footer {

background: rgb(9,18,127);

background: radial-gradient(circle, rgba(9,18,127,1) 0%, rgba(151,208,237,1) 0%);

text-align: center;

padding: 10px;

}

</style>

<asp:ContentPlaceHolder id="head" runat="server">

</asp:ContentPlaceHolder>

</head>

<body>

<form id="form1" runat="server">

<div id="master-container">

<header>

<div class="website-name">

<img src="images/goservice-high-resolution-logo-black-transparent.png" />

</div>

<nav>

<ul>

<li><a href="Default.aspx">Home</a></li>

<li><a href="Services.aspx">Services</a></li>

<li><a href="AboutUs.aspx">About Us</a></li>

<li><a href="Feedback.aspx">Feedback</a></li>

<li><a href="Login.aspx">Login</a></li>

</ul>

</nav>

</header>

<div class="content">

<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

<h2>Welcome to goService!</h2>

<p>This is the default content. Update the content in your specific pages.</p>

</asp:ContentPlaceHolder>

</div>

<footer>

<p>&copy; 2023 goService. All rights reserved.</p>

</footer>

</div>

</form>

</body>

</html>

**:=>** **MasterPage.master.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Xml.Linq;

public partial class MasterPage : System.Web.UI.MasterPage

{

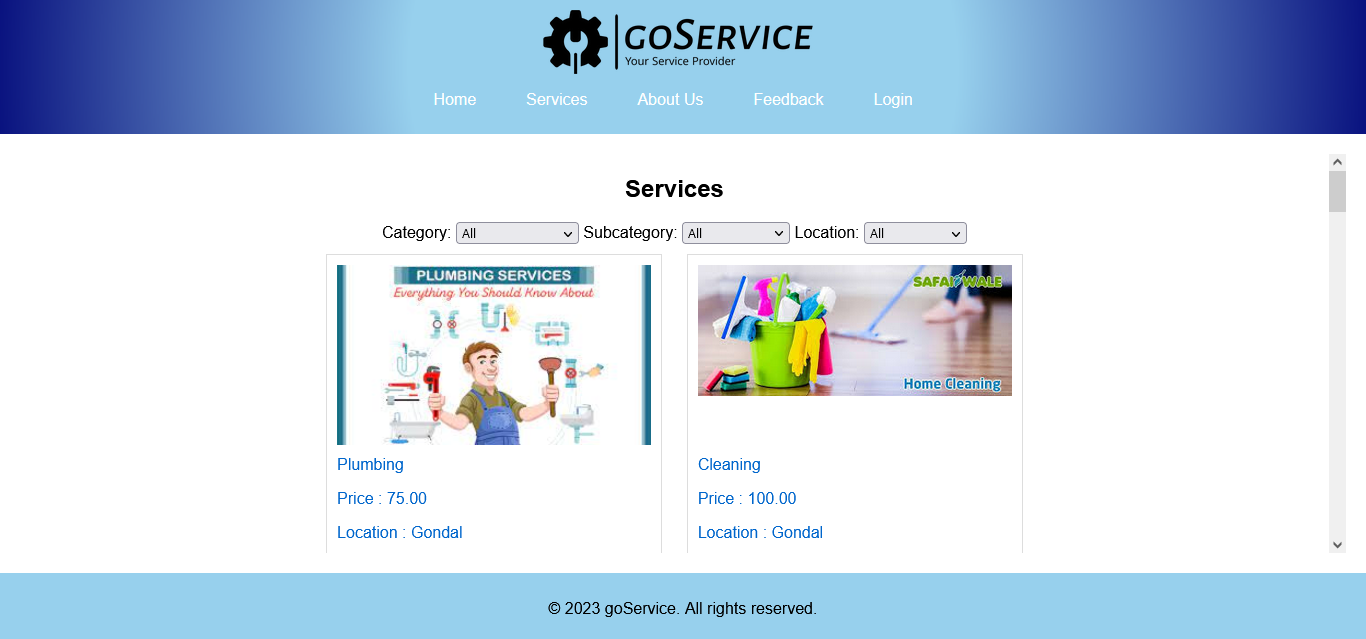
protected void Page\_Load(object sender, EventArgs e)

{

}

}

**Services**

****

**:=>** **Services.aspx**

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Services.aspx.cs" Inherits="Services" Title="Services Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style type="text/css">

body {

text-align: center;

}

.layout {

width: 80%;

margin: auto;

}

.service-square {

display: inline-block;

width: 30%;

height: 300px;

margin: 10px;

padding: 10px;

border: 1px solid #ddd;

text-align: left;

}

.service-square:hover {

box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);

}

.image-container {

width: 100%;

height: 60%;

text-align: center;

}

.service-image {

width: 100%;

height: auto;

max-width: 100%;

max-height: 100%;

}

.service-details {

padding-top: 10px;

}

.service-square a {

text-decoration: none;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<h2>Services</h2>

<div>

<label>Category:</label>

<asp:DropDownList ID="ddlCategory" runat="server" AutoPostBack="True" OnSelectedIndexChanged="ddlCategory\_SelectedIndexChanged">

</asp:DropDownList>

<label>Subcategory:</label>

<asp:DropDownList ID="ddlSubcategory" runat="server" AutoPostBack="True" OnSelectedIndexChanged="ddlSubcategory\_SelectedIndexChanged">

</asp:DropDownList>

<label>Location:</label>

<asp:DropDownList ID="ddlLocation" runat="server" AutoPostBack="True" OnSelectedIndexChanged="ddlLocation\_SelectedIndexChanged">

</asp:DropDownList>

</div>

<asp:Label ID="lblSelectedFilters" runat="server" Text=""></asp:Label>

<section class="layout">

<asp:Repeater ID="RepeaterServices" runat="server">

<ItemTemplate>

<div class="service-square">

<asp:HyperLink ID="hlService" runat="server" NavigateUrl='<%# "ServiceDetails.aspx?ServiceID=" + Eval("ServiceID") %>'>

<div class="image-container">

<asp:Image ID="imgService" runat="server" CssClass="service-image" ImageUrl='<%# ResolveUrl(Eval("ImagePath").ToString()) %>' AlternateText="Service Image" />

</div>

<div class="service-details">

<asp:Label ID="lblServiceName" runat="server" Text='<%# Eval("ServiceName") %>' />

<p>Price : <%# Eval("Price")%></p>

<p> Location : <%# Eval("Location")%></p>

<p>Overall Rating: <%# Eval("OverallRating", "{0:F1}")%></p>

</div>

</asp:HyperLink>

</div>

</ItemTemplate>

</asp:Repeater>

</section>

</asp:Content>

**:=> Services.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using System.Collections.Generic;

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

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

LoadFilterOptions(null);

LoadServices();

}

}

private void LoadAllServices()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT s.ServiceID, s.ServiceName, s.ImagePath, s.Price, s.Location, r.OverallRating " +

"FROM Services s " +

"LEFT JOIN (SELECT ServiceID, AVG(CAST(RatingValue AS FLOAT)) AS OverallRating " +

" FROM Rating GROUP BY ServiceID) r ON s.ServiceID = r.ServiceID " +

"WHERE s.ApprovalStatus = 'Approved'";

using (SqlCommand command = new SqlCommand(query, connection))

{

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader();

RepeaterServices.DataSource = reader;

RepeaterServices.DataBind();

reader.Close();

}

catch (Exception ex)

{

Response.Write("Error: " + ex.Message);

}

}

}

}

private void LoadFilterOptions(string selectedCategory)

{

if (!IsPostBack)

{

LoadDropDownListWithAll("SELECT DISTINCT Category FROM Services", ddlCategory, "Category");

}

if (!string.IsNullOrEmpty(selectedCategory))

{

LoadSubcategories(selectedCategory);

LoadLocations(selectedCategory);

ddlCategory.SelectedValue = selectedCategory;

}

else

{

if (!IsPostBack)

{

LoadDropDownListWithAll("SELECT DISTINCT Subcategory FROM Services", ddlSubcategory, "Subcategory");

LoadDropDownListWithAll("SELECT DISTINCT Location FROM Services", ddlLocation, "Location");

}

}

}

private void LoadDropDownListWithAll(string query, DropDownList ddl, string columnName)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

using (SqlCommand command = new SqlCommand(query, connection))

{

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader();

ddl.Items.Clear();

ddl.Items.Add(new ListItem("All", string.Empty));

while (reader.Read())

{

string value = reader[columnName].ToString();

ListItem item = new ListItem(value, value);

ddl.Items.Add(item);

if (!IsPostBack && ddl.SelectedValue == value)

{

item.Selected = true;

}

}

reader.Close();

}

catch (Exception ex)

{

Response.Write("Error loading filter options: " + ex.Message);

}

}

}

}

private void LoadSubcategories(string selectedCategory)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT DISTINCT Subcategory FROM Services WHERE Category = @SelectedCategory";

SqlCommand cmd = new SqlCommand(query, connection);

cmd.Parameters.AddWithValue("@SelectedCategory", selectedCategory);

try

{

connection.Open();

SqlDataReader reader = cmd.ExecuteReader();

ddlSubcategory.Items.Clear();

ddlSubcategory.Items.Add(new ListItem("All", string.Empty));

while (reader.Read())

{

ddlSubcategory.Items.Add(reader["Subcategory"].ToString());

}

reader.Close();

}

catch (Exception ex)

{

Response.Write("An error occurred while loading subcategories: " + ex.Message);

}

}

}

private void LoadLocations(string selectedCategory)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT DISTINCT Location FROM Services WHERE Category = @SelectedCategory";

SqlCommand cmd = new SqlCommand(query, connection);

cmd.Parameters.AddWithValue("@SelectedCategory", selectedCategory);

try

{

connection.Open();

SqlDataReader reader = cmd.ExecuteReader();

ddlLocation.Items.Clear();

ddlLocation.Items.Add(new ListItem("All", string.Empty));

while (reader.Read())

{

ddlLocation.Items.Add(reader["Location"].ToString());

}

reader.Close();

}

catch (Exception ex)

{

Response.Write("An error occurred while loading locations: " + ex.Message);

}

}

}

protected void ddlCategory\_SelectedIndexChanged(object sender, EventArgs e)

{

LoadFilterOptions(ddlCategory.SelectedValue);

LoadServices();

}

protected void ddlSubcategory\_SelectedIndexChanged(object sender, EventArgs e)

{

LoadServices();

}

protected void ddlLocation\_SelectedIndexChanged(object sender, EventArgs e)

{

LoadServices();

}

private string GetSelectedFiltersText()

{

List<string> selectedFilters = new List<string>();

if (!string.IsNullOrEmpty(ddlCategory.SelectedValue) && ddlCategory.SelectedValue != "All")

{

selectedFilters.Add(" Showing Result For Category: " + ddlCategory.SelectedValue);

}

if (!string.IsNullOrEmpty(ddlSubcategory.SelectedValue) && ddlSubcategory.SelectedValue != "All")

{

selectedFilters.Add("Subcategory: " + ddlSubcategory.SelectedValue);

}

else if (!string.IsNullOrEmpty(ddlCategory.SelectedValue) && ddlSubcategory.SelectedValue == "All")

{

selectedFilters.Add("Category: " + ddlCategory.SelectedValue);

}

if (!string.IsNullOrEmpty(ddlLocation.SelectedValue) && ddlLocation.SelectedValue != "All")

{

selectedFilters.Add("Location: " + ddlLocation.SelectedValue);

}

else if (!string.IsNullOrEmpty(ddlCategory.SelectedValue) && ddlLocation.SelectedValue == "All" && ddlSubcategory.SelectedValue == "All")

{

selectedFilters.Add("Category: " + ddlCategory.SelectedValue);

}

return string.Join(" | ", selectedFilters.ToArray());

}

private void LoadServices()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT s.ServiceID, s.ServiceName, s.ImagePath, s.Price, s.Location, r.OverallRating " +

"FROM Services s " +

"LEFT JOIN (SELECT ServiceID, AVG(CAST(RatingValue AS FLOAT)) AS OverallRating " +

" FROM Rating GROUP BY ServiceID) r ON s.ServiceID = r.ServiceID " +

"WHERE s.ApprovalStatus = 'Approved'";

if (!string.IsNullOrEmpty(ddlCategory.SelectedValue) && ddlCategory.SelectedValue != "All")

{

query += " AND s.Category = @Category";

}

if (!string.IsNullOrEmpty(ddlSubcategory.SelectedValue) && ddlSubcategory.SelectedValue != "All")

{

query += " AND s.Subcategory = @Subcategory";

}

if (!string.IsNullOrEmpty(ddlLocation.SelectedValue) && ddlLocation.SelectedValue != "All")

{

query += " AND s.Location = @Location";

}

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@Category", ddlCategory.SelectedValue);

command.Parameters.AddWithValue("@Subcategory", ddlSubcategory.SelectedValue);

command.Parameters.AddWithValue("@Location", ddlLocation.SelectedValue);

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader();

RepeaterServices.DataSource = reader;

RepeaterServices.DataBind();

reader.Close();

}

catch (Exception ex)

{

Response.Write("Error: " + ex.Message);

}

}

}

string selectedFilters = GetSelectedFiltersText();

lblSelectedFilters.Text = selectedFilters;

}

}

**Service Details**

**:=>** **ServiceDetails.aspx**

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="ServiceDetails.aspx.cs" Inherits="ServiceDetails" Title="Service Details Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style type="text/css">

.service-details-container {

display: flex;

justify-content: space-between;

margin-top: 20px;

}

.service-details {

width: 60%;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

}

.service-details h2 {

color: #333;

}

.ratings-container {

width: 35%;

border: 1px solid #ddd;

border-radius: 8px;

padding: 20px;

overflow-y: auto;

max-height: 400px;

}

.rating-item {

margin-bottom: 20px;

border-bottom: 1px solid #ddd;

padding-bottom: 10px;

}

.feedback-link {

display: block;

margin-top: 20px;

color: #007bff;

text-decoration: none;

}

.feedback-link:hover {

text-decoration: underline;

}

#ctl00\_ContentPlaceHolder1\_lblServiceDetails {

font-family: Arial, sans-serif;

font-size: 25px;

line-height: 1.6;

color: #333;

}

#ctl00\_ContentPlaceHolder1\_lblServiceDetails a {

color: #007bff;

text-decoration: none;

}

#ctl00\_ContentPlaceHolder1\_lblServiceDetails a:hover {

text-decoration: underline;

}

.label {

color: #333;

font-weight: bold;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<h2>Service Details</h2>

<div class="service-details-container">

<div class="service-details">

<asp:Label ID="lblServiceDetails" runat="server" Text=""></asp:Label>

</div>

<div class="ratings-container">

<a class="feedback-link" href='<%= ResolveUrl("~/GiveRating.aspx?ServiceID=" + Request.QueryString["ServiceID"]) %>' target="\_blank">Give Feedback</a>

<asp:Repeater ID="RepeaterRatings" runat="server">

<ItemTemplate>

<div class="rating-item">

<p><strong>Username:</strong> <%# Eval("Username") %></p>

<p><strong>Rating:</strong> <%# Eval("RatingValue") %></p>

<p><strong>Additional Info:</strong> <%# Eval("AdditionalInfo") %></p>

</div>

</ItemTemplate>

</asp:Repeater>

</div>

</div>

</asp:Content>

**:=>** **ServiceDetails.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

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

{

private string connectionString = System.Configuration.ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

LoadServiceDetails();

LoadRatings();

}

}

private void LoadServiceDetails()

{

if (Request.QueryString["ServiceID"] != null)

{

int serviceID = Convert.ToInt32(Request.QueryString["ServiceID"]);

using (SqlConnection connection = new SqlConnection(connectionString))

{

string serviceQuery = "SELECT s.ServiceName, s.Description, s.Category, s.Subcategory, s.Location, s.Price, u.UserName as ProviderName, s.ProviderID " +

"FROM Services s " +

"INNER JOIN [User] u ON s.ProviderID = u.UserID " +

"WHERE s.ServiceID = @ServiceID";

SqlCommand serviceCommand = new SqlCommand(serviceQuery, connection);

serviceCommand.Parameters.AddWithValue("@ServiceID", serviceID);

string ratingQuery = "SELECT RatingValue FROM Rating WHERE ServiceID = @ServiceID";

SqlCommand ratingCommand = new SqlCommand(ratingQuery, connection);

ratingCommand.Parameters.AddWithValue("@ServiceID", serviceID);

try

{

connection.Open();

SqlDataReader serviceReader = serviceCommand.ExecuteReader();

if (serviceReader.Read())

{

string serviceName = serviceReader["ServiceName"].ToString();

string description = serviceReader["Description"].ToString();

string category = serviceReader["Category"].ToString();

string subcategory = serviceReader["Subcategory"].ToString();

string location = serviceReader["Location"].ToString();

decimal price = Convert.ToDecimal(serviceReader["Price"]);

string providerName = serviceReader["ProviderName"].ToString();

int providerID = Convert.ToInt32(serviceReader["ProviderID"]);

lblServiceDetails.Text = "<span class='label'>Service Name:</span> " + serviceName + "<br />" +

"<span class='label'>Description:</span> " + description + "<br />" +

"<span class='label'>Category:</span> " + category + "<br />" +

"<span class='label'>Subcategory:</span> " + subcategory + "<br />" +

"<span class='label'>Location:</span> " + location + "<br />" +

"<span class='label'>Price:</span> " + price + "<br />" +

"<span class='label'>Provider:</span> <a href='ProviderProfile.aspx?ProviderID=" + providerID + "'>" + providerName + "</a><br />";

serviceReader.Close();

}

SqlDataReader ratingReader = ratingCommand.ExecuteReader();

int totalRating = 0;

int ratingCount = 0;

while (ratingReader.Read())

{

totalRating += Convert.ToInt32(ratingReader["RatingValue"]);

ratingCount++;

}

ratingReader.Close();

if (ratingCount > 0)

{

double overallRating = (double)totalRating / ratingCount;

lblServiceDetails.Text += "Overall Rating: " + overallRating.ToString("F1");

}

}

catch (Exception ex)

{

lblServiceDetails.Text = "Error loading service details.";

Response.Write("Error: " + ex.Message);

}

}

}

}

private void LoadRatings()

{

if (Request.QueryString["ServiceID"] != null)

{

int serviceID = Convert.ToInt32(Request.QueryString["ServiceID"]);

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT UserName, RatingValue, AdditionalInfo FROM Rating WHERE ServiceID = @ServiceID";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@ServiceID", serviceID);

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader();

RepeaterRatings.DataSource = reader;

RepeaterRatings.DataBind();

reader.Close();

}

catch (Exception ex)

{

Response.Write("Error: " + ex.Message);

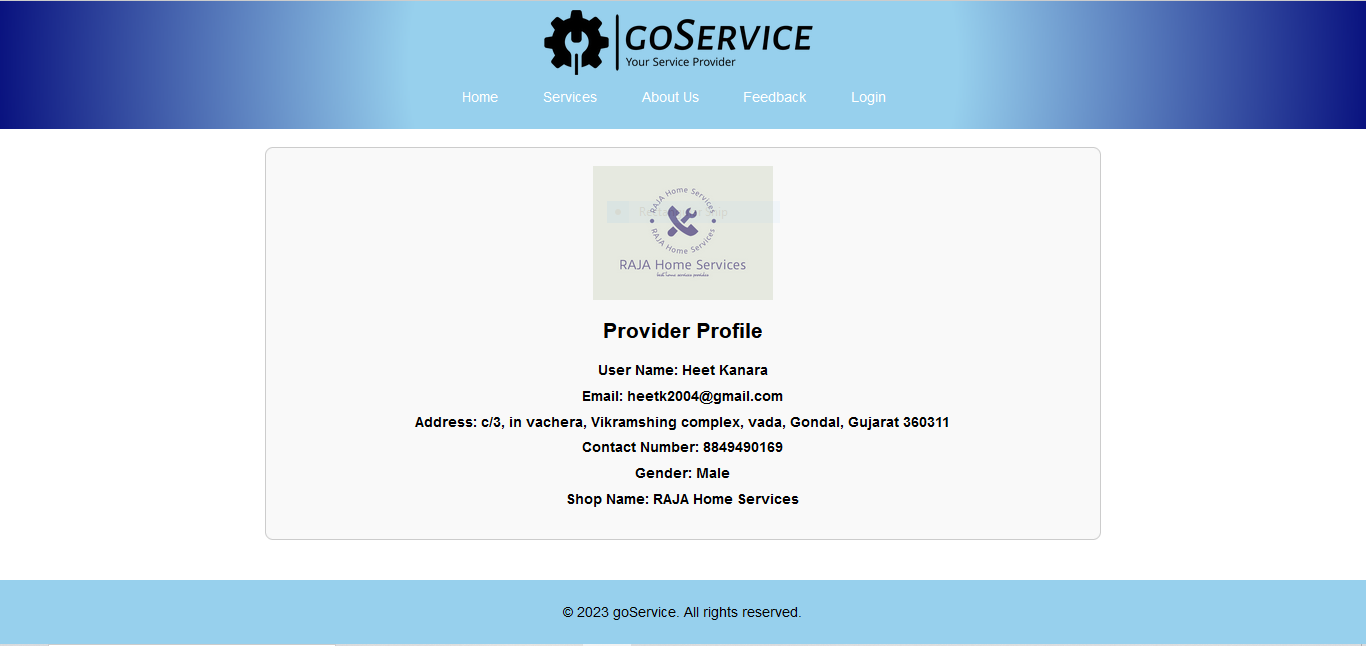
}

}

}

}

}

**Provider Profile**

**:=> ProviderProfile.aspx**

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="ProviderProfile.aspx.cs" Inherits="ProviderProfile" Title="Provider Profile Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

.provider-profile {

width: 60%;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 8px;

background-color: #f9f9f9;

}

.profile-heading {

font-size: 24px;

font-weight: bold;

margin-bottom: 20px;

}

.profile-image {

max-width: 200px;

max-height: 200px;

margin: 0 auto 20px;

display: block;

}

.profile-details {

margin-bottom: 15px;

}

.profile-label {

font-weight: bold;

}

.profile-info {

margin-bottom: 10px;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div class="provider-profile">

<img class="profile-image" id="imgProvider" runat="server" alt="Provider Image" />

<h2 class="profile-heading">Provider Profile</h2>

<div class="profile-details">

<div class="profile-info">

<asp:Label ID="lblProviderName" runat="server" CssClass="profile-label"></asp:Label>

</div>

<div class="profile-info">

<asp:Label ID="lblEmail" runat="server" CssClass="profile-label"></asp:Label>

</div>

<div class="profile-info">

<asp:Label ID="lblAddress" runat="server" CssClass="profile-label"></asp:Label>

</div>

<div class="profile-info">

<asp:Label ID="lblContactNumber" runat="server" CssClass="profile-label"></asp:Label>

</div>

<div class="profile-info">

<asp:Label ID="lblGender" runat="server" CssClass="profile-label"></asp:Label>

</div>

<div class="profile-info">

<asp:Label ID="lblShopName" runat="server" CssClass="profile-label"></asp:Label>

</div>

</div>

</div>

</asp:Content>

**:=>** **ProviderProfile.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

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

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

LoadProviderDetails();

}

}

private void LoadProviderDetails()

{

if (Request.QueryString["ProviderID"] != null)

{

int providerID = Convert.ToInt32(Request.QueryString["ProviderID"]);

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT UserName, Email, Address, ContactNumber, Gender, ShopName, Image FROM [User] WHERE UserID = @ProviderID";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@ProviderID", providerID);

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

lblProviderDetails.Text = "User Name: " + reader["UserName"] + "<br />" +

"Email: " + reader["Email"] + "<br />" +

"Address: " + reader["Address"] + "<br />" +

"Contact Number: " + reader["ContactNumber"] + "<br />" +

"Gender: " + reader["Gender"] + "<br />" +

"Shop Name: " + reader["ShopName"] + "<br />";

if (!DBNull.Value.Equals(reader["Image"]))

{

string imagePath = reader["Image"].ToString();

lblProviderDetails.Text += "Image: <img src='" + ResolveUrl(imagePath) + "' alt='Provider Image' width='100' height='100' /><br />";

}

else

{

lblProviderDetails.Text += "Image: Not available<br />";

}

}

reader.Close();

}

catch (Exception ex)

{

lblProviderDetails.Text = "Error loading provider details: " + ex.Message;

}

}

}

}

else

{

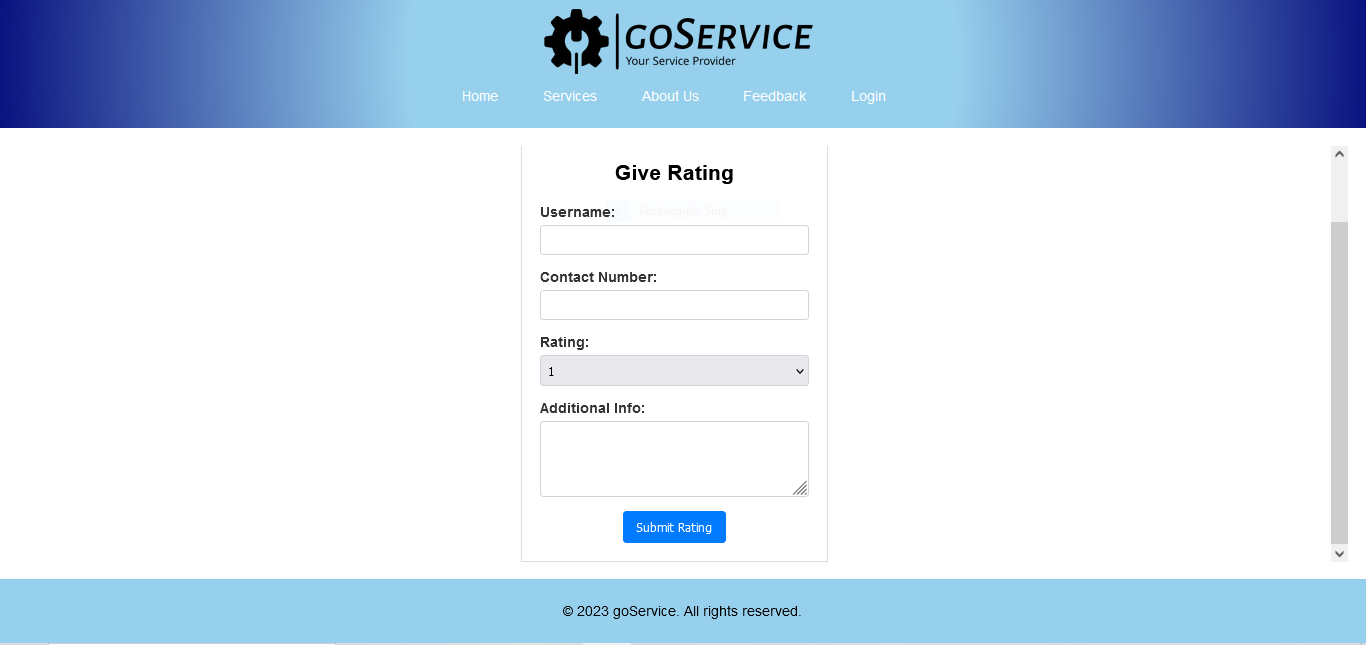
lblProviderDetails.Text = "Provider ID is missing in the query string.";

}

}

}

**Give Rating**

****

**:=>** **GiveRating.aspx**

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="GiveRating.aspx.cs" Inherits="GiveRating" Title="Give Rating Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<title>Give Rating</title>

<style type="text/css">

.rating-form {

width: 300px;

margin: 0 auto;

text-align: center;

padding: 20px;

border: 1px solid #ddd;

margin-top: 50px;

}

.form-label {

display: block;

font-weight: bold;

margin-bottom: 5px;

text-align: left;

color: #333;

}

.form-group {

margin-bottom: 15px;

}

.form-control {

width: 100%;

padding: 8px;

box-sizing: border-box;

border: 1px solid #ced4da;

border-radius: 4px;

}

.form-control:focus {

outline: none;

border-color: #007bff;

box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);

}

.btnLogin {

background-color: #007bff;

color: white;

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

}

.btnLogin:hover {

background-color: #0056b3;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div class="rating-form">

<h2>Give Rating</h2>

<div class="form-group">

<asp:Label ID="lblUsername" runat="server" Text="Username:" CssClass="form-label"></asp:Label>

<asp:TextBox ID="txtUsername" runat="server" MaxLength="15" CssClass="form-control"></asp:TextBox>

</div>

<div class="form-group">

<asp:Label ID="lblContactNumber" runat="server" Text="Contact Number:" CssClass="form-label"></asp:Label>

<asp:TextBox ID="txtContactNumber" runat="server" MaxLength="13" CssClass="form-control"></asp:TextBox>

<asp:RegularExpressionValidator ID="revContactNumber" runat="server"

ControlToValidate="txtContactNumber"

ErrorMessage="Enter a valid contact number"

ValidationExpression="^\d{10}$"

Display="Dynamic">

</asp:RegularExpressionValidator>

</div>

<div class="form-group">

<asp:Label ID="lblRating" runat="server" Text="Rating:" CssClass="form-label"></asp:Label>

<asp:DropDownList ID="ddlRating" runat="server" CssClass="form-control">

<asp:ListItem Text="1" Value="1"></asp:ListItem>

<asp:ListItem Text="2" Value="2"></asp:ListItem>

<asp:ListItem Text="3" Value="3"></asp:ListItem>

<asp:ListItem Text="4" Value="4"></asp:ListItem>

<asp:ListItem Text="5" Value="5"></asp:ListItem>

</asp:DropDownList>

</div>

<div class="form-group">

<asp:Label ID="lblAdditionalInfo" runat="server" Text="Additional Info:" CssClass="form-label"></asp:Label>

<asp:TextBox ID="txtAdditionalInfo" runat="server" TextMode="MultiLine"

Rows="3" MaxLength="50" CssClass="form-control"></asp:TextBox>

</div>

<asp:Button ID="btnSubmitRating" runat="server" Text="Submit Rating" OnClick="btnSubmitRating\_Click" CssClass="btnLogin" />

</div>

</asp:Content>

**:=>** **GiveRating.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

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

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

}

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

{

string username = txtUsername.Text;

string contactNumber = txtContactNumber.Text;

int ratingValue = Convert.ToInt32(ddlRating.SelectedValue);

string additionalInfo = txtAdditionalInfo.Text;

if (!string.IsNullOrEmpty(Request.QueryString["ServiceID"]))

{

int serviceID = Convert.ToInt32(Request.QueryString["ServiceID"]);

int providerID = GetProviderIDForService(serviceID);

InsertRating(username, contactNumber, ratingValue, additionalInfo, serviceID, providerID);

Response.Redirect("ServiceDetails.aspx?ServiceID=" + serviceID);

}

else

{

}

}

private int GetProviderIDForService(int serviceID)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT ProviderID FROM Services WHERE ServiceID = @ServiceID";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@ServiceID", serviceID);

connection.Open();

object result = command.ExecuteScalar();

if (result != null)

{

return Convert.ToInt32(result);

}

else

{

return -1;

}

}

}

}

private void InsertRating(string username, string contactNumber, int ratingValue, string additionalInfo, int serviceID, int providerID)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "INSERT INTO Rating (UserName, ContactNumber, RatingValue, AdditionalInfo, ServiceID, ProviderID) " +

"VALUES (@UserName, @ContactNumber, @RatingValue, @AdditionalInfo, @ServiceID, @ProviderID)";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@UserName", username);

command.Parameters.AddWithValue("@ContactNumber", contactNumber);

command.Parameters.AddWithValue("@RatingValue", ratingValue);

command.Parameters.AddWithValue("@AdditionalInfo", additionalInfo);

command.Parameters.AddWithValue("@ServiceID", serviceID);

command.Parameters.AddWithValue("@ProviderID", providerID);

connection.Open();

command.ExecuteNonQuery();

}

}

}

}

**About Us**



**:=>** **AboutUs.aspx**

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="AboutUs.aspx.cs" Inherits="AboutUs" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

.container {

width: 50%;

margin: auto;

padding: 20px;

background-color: #fff;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

text-align: center;

font-family: 'Arial', sans-serif;

}

h2 {

color: #3498db;

font-size: 2em;

p {

line-height: 1.6;

font-size: 1.2em;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div class="container">

<h2>Who we are</h2>

<p>

goService is a technology platform offering a variety of services at home. Customers use our platform to book services such as beauty treatments, haircuts, massage therapy, cleaning, plumbing, carpentry, appliance repair, painting, etc. These services are delivered in the comfort of their home and at a time of their choosing. We promise our customers a high quality, standardized, and reliable service experience. To fulfill this promise, we work closely with our hand-picked service partners, enabling them with technology, training, products, tools, financing, insurance, and brand, helping them succeed and deliver on this promise.

</p>

<p>

<strong>Our Vision:</strong> Empower millions of professionals worldwide to deliver services at home like never experienced before.

</p>

</div>

</asp:Content>

**:=>** **AboutUs.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

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

{

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

{

}

}

**Feedback**



**:=>** **Feedback.aspx**

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Feedback.aspx.cs" Inherits="Feedback" Title="Feedback" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div>

<h2>Feedback</h2>

<asp:Label ID="lblMessage" runat="server" ForeColor="Green" Visible="false"></asp:Label>

<br />

<asp:Label ID="lblError" runat="server" ForeColor="Red" Visible="false"></asp:Label>

<br />

<asp:TextBox ID="txtUserName" runat="server" placeholder="Your Name" Width="300" CssClass="form-control"></asp:TextBox>

<br />

<asp:TextBox ID="txtContactNumber" runat="server" placeholder="Contact Number" Width="300" CssClass="form-control"></asp:TextBox>

<br />

<asp:TextBox ID="txtFeedback" runat="server" TextMode="MultiLine" Rows="4" placeholder="Your Feedback" Width="300" CssClass="form-control"></asp:TextBox>

<br />

<asp:Button ID="btnSubmit" runat="server" Text="Submit Feedback" OnClick="btnSubmit\_Click" CssClass="btn btn-primary" />

</div>

</asp:Content>

**:=>** **Feedback.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

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

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

}

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

{

string userName = txtUserName.Text.Trim();

string contactNumber = txtContactNumber.Text.Trim();

string feedbackText = txtFeedback.Text.Trim();

if (!string.IsNullOrEmpty(userName) && !string.IsNullOrEmpty(feedbackText))

{

if (SaveFeedback(userName, contactNumber, feedbackText))

{

lblMessage.Text = "Feedback submitted successfully!";

lblMessage.Visible = true;

}

else

{

lblError.Text = "Error submitting feedback. Please try again later.";

lblError.Visible = true;

}

}

else

{

lblError.Text = "Name and Feedback are required fields.";

lblError.Visible = true;

}

}

private bool SaveFeedback(string userName, string contactNumber, string feedbackText)

{

try

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "INSERT INTO Feedback (UserName, ContactNumber, FeedbackText, DateSubmitted) VALUES (@UserName, @ContactNumber, @FeedbackText, GETDATE())";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@UserName", userName);

command.Parameters.AddWithValue("@ContactNumber", contactNumber);

command.Parameters.AddWithValue("@FeedbackText", feedbackText);

connection.Open();

command.ExecuteNonQuery();

return true;

}

}

}

catch (Exception ex)

{

Console.WriteLine("Exception: " + ex.Message);

return false;

}

}

}

**Login**



**:=> Login.aspx**

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Login.aspx.cs" Inherits="Login" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

<style type="text/css">

.page-container {

display: flex;

align-items: center;

justify-content: center;

margin-top: 50px;

}

.login-container {

background-color: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 600px;

box-sizing: border-box;

text-align: center;

}

.login-container label {

display: block;

font-weight: bold;

margin-bottom: 5px;

text-align: left;

}

.login-container .form-control {

width: 100%;

padding: 8px;

box-sizing: border-box;

border: 1px solid #ced4da;

border-radius: 4px;

margin-bottom: 10px;

}

.login-container .form-control:focus {

outline: none;

border-color: #007bff;

box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);

}

.login-container .btnLogin {

background-color: #007bff;

color: white;

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

}

.login-container .btnLogin:hover {

background-color: #0056b3;

}

.login-container p {

margin-top: 15px;

}

.login-container a {

color: #007bff;

text-decoration: none;

font-weight: bold;

}

.login-container a:hover {

text-decoration: underline;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<div class="page-container">

<div class="login-container">

<label for="txtLoginEmail">Email:</label>

<asp:TextBox ID="txtLoginEmail" runat="server" CssClass="form-control" />

<label for="txtLoginPassword">Password:</label>

<asp:TextBox ID="txtLoginPassword" runat="server" TextMode="Password"

CssClass="form-control" MaxLength="10" />

<asp:Button ID="btnLogin" runat="server" Text="Login" OnClick="btnLogin\_Click" CssClass="btnLogin" />

<p>Don't have an account? <a href="Register.aspx">Register here</a>.</p>

</div>

</div>

</asp:Content>

**:=> Login.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using System.Security.Cryptography;

using System.Text;

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

{

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

{

string email = txtLoginEmail.Text;

string password = HashPassword(txtLoginPassword.Text);

string connectionString = System.Configuration.ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

using (SqlCommand checkUser = new SqlCommand("SELECT UserID, UserName, UserRole, ApprovalStatus FROM [User] WHERE Email = @Email AND Password = @Password", connection))

{

checkUser.Parameters.AddWithValue("@Email", email);

checkUser.Parameters.AddWithValue("@Password", password);

using (SqlDataReader reader = checkUser.ExecuteReader())

{

if (reader.Read())

{

string approvalStatus = reader["ApprovalStatus"].ToString();

if (approvalStatus == "Approved")

{

int userId = Convert.ToInt32(reader["UserID"]);

string userRole = reader["UserRole"].ToString();

string userName = reader["UserName"].ToString();

Session["UserID"] = userId;

Session["UserName"] = userName;

if (userRole == "Admin")

{

Response.Redirect("Admin/AdminHome.aspx");

}

else

{

Response.Redirect("User/UserHome.aspx");

}

}

else if (approvalStatus == "Pending")

{

ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Your account is pending approval. Please wait for approval.');", true);

}

else if (approvalStatus == "Rejected")

{

ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Your account has been rejected. Please contact support.');", true);

}

}

else

{

ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Invalid email or password');", true);

}

}

}

}

}

private string HashPassword(string password)

{

using (SHA256 sha256 = SHA256.Create())

{

byte[] hashedBytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));

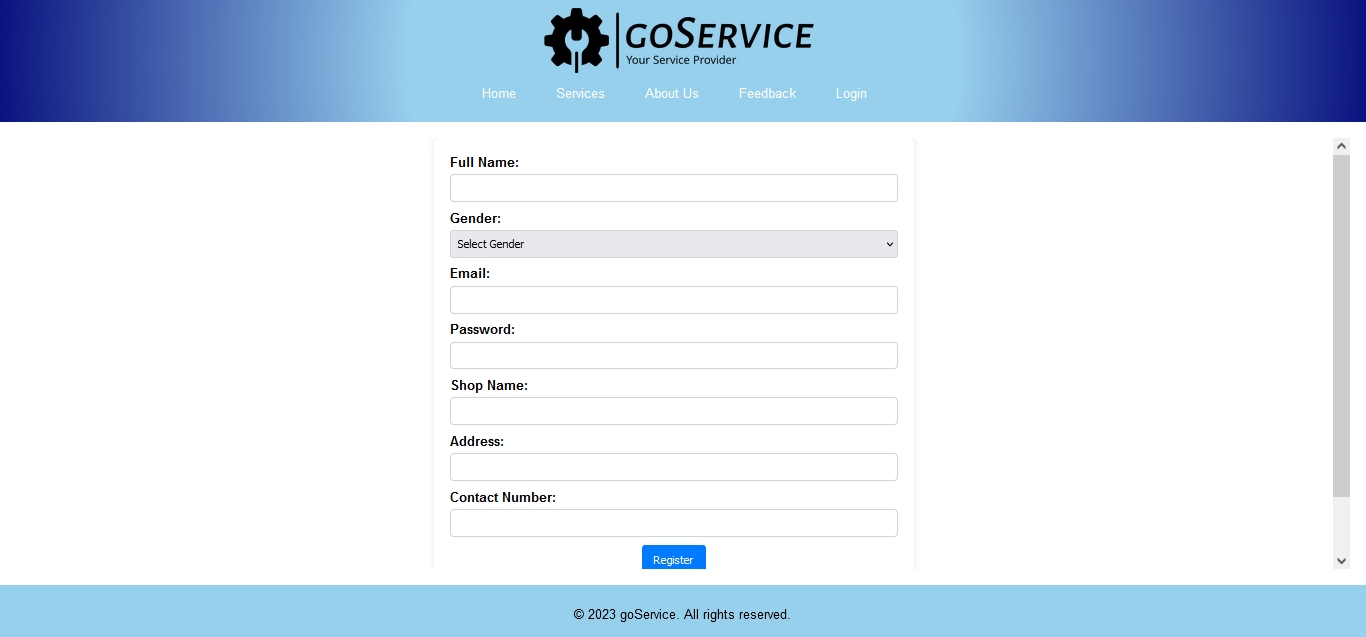
return BitConverter.ToString(hashedBytes).Replace("-", "").ToLower();

}

}

}

**Register**



**:=> Register.aspx**

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Register.aspx.cs" Inherits="Register" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

<style type="text/css">

.page-container {

display: flex;

align-items: center;

justify-content: center;

margin: 0;

}

.register-container {

background-color: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 600px;

box-sizing: border-box;

text-align: center;

}

.register-container label {

display: block;

font-weight: bold;

margin-bottom: 5px;

text-align: left;

}

.register-container .form-control {

width: 100%;

padding: 8px;

box-sizing: border-box;

border: 1px solid #ced4da;

border-radius: 4px;

margin-bottom: 10px;

}

.register-container .btnRegister {

background-color: #007bff;

color: white;

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

}

.register-container .btnRegister:hover {

background-color: #0056b3;

}

.register-container p {

margin-top: 15px;

}

.register-container a {

color: #007bff;

text-decoration: none;

font-weight: bold;

}

.register-container a:hover {

text-decoration: underline;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<div class="page-container">

<div class="register-container">

<label for="txtFullName">Full Name:</label>

<asp:TextBox ID="txtFullName" runat="server" CssClass="form-control"

MaxLength="15" />

<label for="ddlGender">Gender:</label>

<asp:DropDownList ID="ddlGender" runat="server" CssClass="form-control">

<asp:ListItem Text="Select Gender" Value="" />

<asp:ListItem Text="Male" Value="Male" />

<asp:ListItem Text="Female" Value="Female" />

</asp:DropDownList>

<label for="txtEmail">Email:</label>

<asp:TextBox ID="txtEmail" runat="server" CssClass="form-control" />

<asp:RegularExpressionValidator ID="regexEmail" runat="server"

ControlToValidate="txtEmail"

ValidationExpression="^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"

ErrorMessage="Enter a valid email address."

Display="Dynamic"

CssClass="text-danger" />

<label for="txtPassword">Password:</label>

<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"

CssClass="form-control" MaxLength="10" />

<label for="txtShopName">Shop Name:</label>

<asp:TextBox ID="txtShopName" runat="server" CssClass="form-control"

MaxLength="15" />

<label for="txtAddress">Address:</label>

<asp:TextBox ID="txtAddress" runat="server" CssClass="form-control"

MaxLength="30" />

<label for="txtContactNumber">Contact Number:</label>

<asp:TextBox ID="txtContactNumber" runat="server" CssClass="form-control"

MaxLength="13" />

<asp:RegularExpressionValidator ID="regexContactNumber" runat="server"

ControlToValidate="txtContactNumber"

ValidationExpression="^\d{10}$"

ErrorMessage="Enter a valid 10-digit contact number."

Display="Dynamic"

CssClass="text-danger" />

<asp:Button ID="btnRegister" runat="server" Text="Register" OnClick="btnRegister\_Click" CssClass="btnRegister" />

<p>Already have an account? <a href="Login.aspx">Login here</a>.</p>

</div>

</div>

</asp:Content>

**:=> Register.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using System.Security.Cryptography;

using System.Text;

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

{

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

{

string fullName = txtFullName.Text;

string gender = ddlGender.SelectedValue;

string email = txtEmail.Text;

string password = HashPassword(txtPassword.Text);

string shopName = txtShopName.Text;

string address = txtAddress.Text;

string contactNumber = txtContactNumber.Text;

string connectionString = System.Configuration.ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand checkUserExistence = new SqlCommand("SELECT COUNT(\*) FROM [User] WHERE Email = @Email", connection);

checkUserExistence.Parameters.AddWithValue("@Email", email);

int userCount = (int)checkUserExistence.ExecuteScalar();

if (userCount > 0)

{

ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Email already exists');", true);

return;

}

SqlCommand insertUser = new SqlCommand(

"INSERT INTO [User] (UserName, Email, Password, UserRole, ShopName, Address, ContactNumber, Gender, ApprovalStatus) " +

"VALUES (@UserName, @Email, @Password, @UserRole, @ShopName, @Address, @ContactNumber, @Gender, 'Pending')",

connection);

insertUser.Parameters.AddWithValue("@UserName", fullName);

insertUser.Parameters.AddWithValue("@Email", email);

insertUser.Parameters.AddWithValue("@Password", password);

insertUser.Parameters.AddWithValue("@UserRole", "User");

insertUser.Parameters.AddWithValue("@ShopName", shopName);

insertUser.Parameters.AddWithValue("@Address", address);

insertUser.Parameters.AddWithValue("@ContactNumber", contactNumber);

insertUser.Parameters.AddWithValue("@Gender", gender);

insertUser.ExecuteNonQuery();

}

ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Registration request submitted. Please wait for admin approval.');", true);

}

private string HashPassword(string password)

{

using (SHA256 sha256 = SHA256.Create())

{

byte[] hashedBytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));

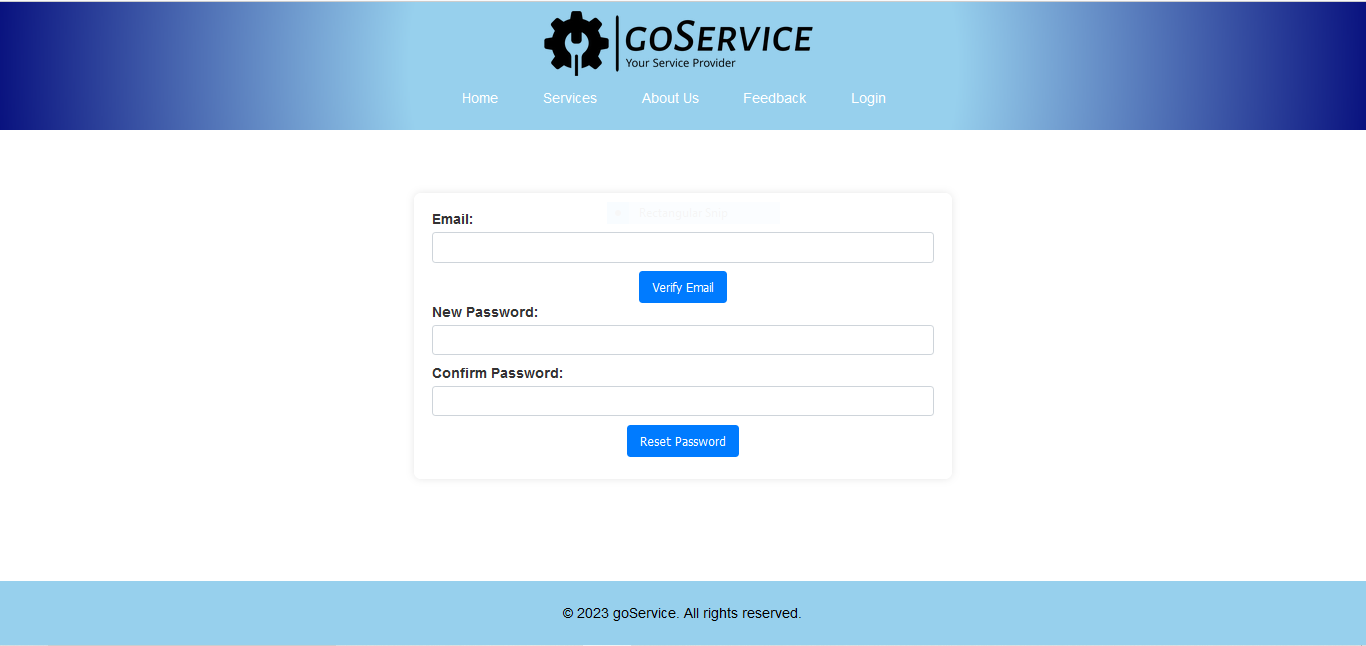
return BitConverter.ToString(hashedBytes).Replace("-", "").ToLower();

}

}

}

**Reset Password**



**:=> ResetPassword.aspx**

<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="ResetPassword.aspx.cs" Inherits="ResetPassword" Title="Reset Password" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style type="text/css">

.page-container {

display: flex;

align-items: center;

justify-content: center;

margin-top: 50px;

}

.login-container {

background-color: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 600px;

box-sizing: border-box;

text-align: center;

}

.label {

display: block;

font-weight: bold;

margin-bottom: 5px;

text-align: left;

color: #333;

}

.login-container .form-control {

width: 100%;

padding: 8px;

box-sizing: border-box;

border: 1px solid #ced4da;

border-radius: 4px;

margin-bottom: 10px;

}

.login-container .form-control:focus {

outline: none;

border-color: #007bff;

box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);

}

.login-container .btnLogin {

background-color: #007bff;

color: white;

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

}

.login-container .btnLogin:hover {

background-color: #0056b3;

}

.login-container p {

margin-top: 15px;

color: #333; }

.login-container a {

color: #007bff;

text-decoration: none;

font-weight: bold;

}

.login-container a:hover {

text-decoration: underline;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div class="page-container">

<div class="login-container">

<asp:Label ID="lblEmail" runat="server" Text="Email:" CssClass="label"></asp:Label>

<asp:TextBox ID="txtEmail" runat="server" CssClass="form-control"></asp:TextBox>

<asp:Button ID="btnVerifyEmail" runat="server" Text="Verify Email" OnClick="btnVerifyEmail\_Click" CssClass="btnLogin" />

<asp:Panel ID="pnlResetPassword" runat="server" Visible="false">

<asp:Label ID="lblNewPassword" runat="server" Text="New Password:" CssClass="label"></asp:Label>

<asp:TextBox ID="txtNewPassword" TextMode="Password" runat="server" CssClass="form-control"></asp:TextBox>

<asp:Label ID="lblConfirmPassword" runat="server" Text="Confirm Password:" CssClass="label"></asp:Label>

<asp:TextBox ID="txtConfirmPassword" TextMode="Password" runat="server" CssClass="form-control"></asp:TextBox>

<asp:Button ID="btnResetPassword" runat="server" Text="Reset Password" OnClick="btnResetPassword\_Click" CssClass="btnLogin" />

</asp:Panel>

<asp:Label ID="lblMessage" runat="server" Text="" CssClass="label"></asp:Label>

</div>

</div>

</asp:Content>

**:=> ResetPassword.aspx**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using System.Security.Cryptography;

using System.Text;

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

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

pnlResetPassword.Visible = false;

}

}

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

{

string email = txtEmail.Text;

bool emailExists = CheckEmailExists(email);

if (emailExists)

{

pnlResetPassword.Visible = true;

lblMessage.Text = "";

}

else

{

lblMessage.Text = "Email does not exist.";

}

}

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

{

string email = txtEmail.Text;

string newPassword = txtNewPassword.Text;

bool passwordReset = ResetPasswordByEmail(email, newPassword);

if (passwordReset)

{

lblMessage.Text = "Password reset successfully.";

}

else

{

lblMessage.Text = "Failed to reset password.";

}

}

private bool CheckEmailExists(string email)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT COUNT(\*) FROM [User] WHERE Email = @Email";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@Email", email);

connection.Open();

int count = (int)command.ExecuteScalar();

return count > 0;

}

}

private bool ResetPasswordByEmail(string email, string newPassword)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "UPDATE [User] SET Password = @NewPassword WHERE Email = @Email";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@NewPassword", HashPassword(newPassword));

command.Parameters.AddWithValue("@Email", email);

connection.Open();

int rowsAffected = command.ExecuteNonQuery();

return rowsAffected > 0;

}

}

private string HashPassword(string password)

{

using (SHA256 sha256 = SHA256.Create())

{

byte[] hashedBytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));

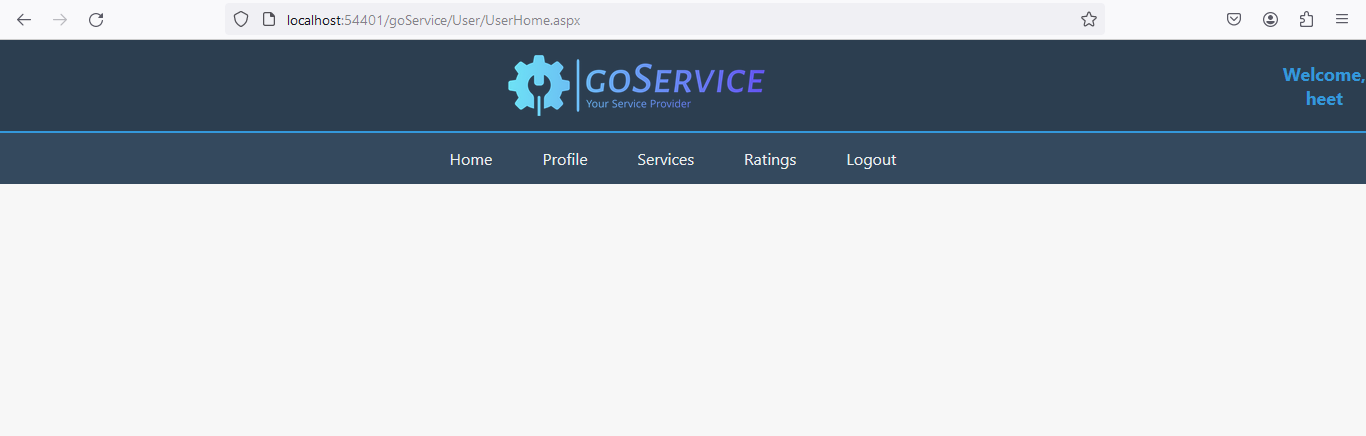
return BitConverter.ToString(hashedBytes).Replace("-", "").ToLower();

}

}

}

**SERVICE PROVIDER SIDE**

**Home Page**

**:=> UserHome.aspx**

<%@ Page Language="C#" MasterPageFile="~/User/UserMasterPage.master" AutoEventWireup="true" CodeFile="UserHome.aspx.cs" Inherits="User\_home" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

</asp:Content>

**:=> UserHome.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

public partial class User\_home : System.Web.UI.Page

{

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

{

}

}

**:=> UserMasterPage.aspx**

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="UserMasterPage.master.cs" Inherits="User\_UserMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head id="Head1" runat="server">

<title>goService</title>

<style>

body {

margin: 0;

padding: 0;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background-color: #f7f7f7;

}

header {

background: #2c3e50;

color: #ecf0f1;

padding: 15px 0;

text-align: center;

}

#master-container {

display: flex;

flex-direction: column;

height: 100vh;

}

.website-name {

display: flex;

align-items: center;

justify-content: center;

}

.website-name h1 {

margin: 0;

font-size: 24px;

color: #3498db;

}

.logo img {

width: 20%;

margin-right: 10px;

}

.welcome-message {

font-size: 18px;

color: #3498db;

margin-top: 10px;

font-weight: bold;

}

nav {

text-align: center;

background-color: #34495e;

border-top: 2px solid #3498db;

}

nav ul {

list-style: none;

padding: 0;

margin: 0;

display: flex;

justify-content: center;

align-items: center;

}

nav ul li {

margin-right: 20px;

}

nav a {

text-decoration: none;

color: #ecf0f1;

padding: 15px;

display: inline-block;

transition: background-color 0.3s, color 0.3s;

}

.content {

text-align: center;

margin: 20px;

flex: 1;

overflow-y: auto;

}

nav a:hover {

background-color: #3498db;

color: #fff;

}

</style>

<asp:ContentPlaceHolder id="head" runat="server">

</asp:ContentPlaceHolder>

</head>

<body>

<form id="form1" runat="server">

<div id="master-container">

<header>

<div class="website-name">

<div class="logo">

<img src="../images/goservice-high-resolution-logo-transparent.png" />

</div>

<div class="user-welcome">

<asp:Label ID="lblWelcomeMessage" runat="server" CssClass="welcome-message"></asp:Label>

</div>

</div>

</header>

<nav>

<ul>

<li><a href="UserHome.aspx">Home</a></li>

<li><a href="UserProfile.aspx">Profile</a></li>

<li><a href="UserServices.aspx">Services</a></li>

<li><a href="UserRating.aspx">Ratings</a></li>

<li><a href="UserLogout.aspx">Logout</a></li>

</ul>

</nav>

<div class="content">

<asp:Label ID="Label1" runat="server" CssClass="welcome-message"></asp:Label>

<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

</asp:ContentPlaceHolder>

</div>

</div>

</form>

</body>

</html>

**:=> UserMasterPage.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Xml.Linq;

public partial class User\_UserMasterPage : System.Web.UI.MasterPage

{

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

{

if (Session["UserID"] != null && Session["UserName"] != null)

{

int userId = Convert.ToInt32(Session["UserID"]);

string userName = Session["UserName"].ToString();

lblWelcomeMessage.Text = "Welcome, " + userName;

}

else

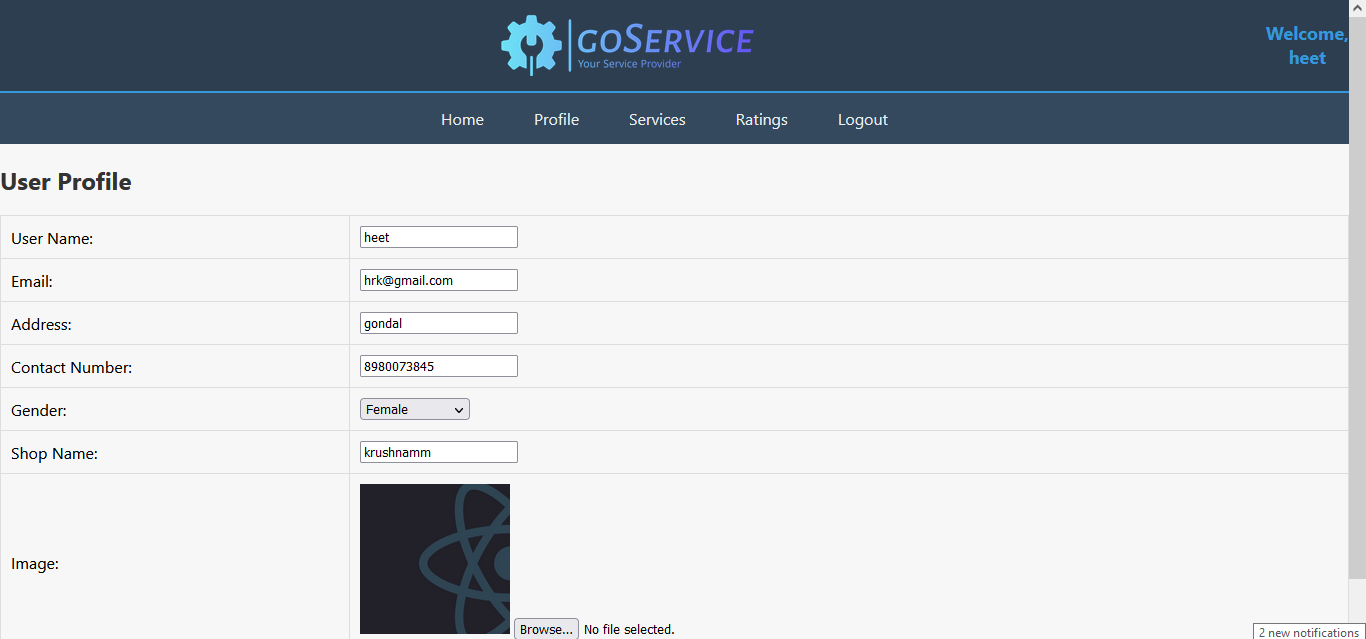
{

Response.Redirect("../Login.aspx");

}

}

}

**User Profile**

**:=> UserProfile.aspx**

<%@ Page Language="C#" MasterPageFile="~/User/UserMasterPage.master" AutoEventWireup="true" CodeFile="UserProfile.aspx.cs" Inherits="User\_UserProfile" Title="User Profile" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

h2 {

color: #333;

}

table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

table, th, td {

border: 1px solid #ddd;

}

th, td {

padding: 10px;

text-align: left;

}

.form-group {

margin-bottom: 15px;

}

label {

display: inline-block;

width: 150px;

font-weight: bold;

margin-right: 10px;

}

.btnEdit,

.btnUpdate {

background-color: #007bff;

color: white;

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

}

.btnUpdate {

background-color: #28a745;

}

.file-upload-container {

margin-top: 15px;

}

#ctl00\_ContentPlaceHolder1\_imgUser {

width: 150px;

height: 150px;

object-fit: cover;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<h2>User Profile</h2>

<div>

<table>

<tr>

<td>

<asp:Label ID="lblUsername" runat="server" Text="User Name:"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtUsername" runat="server" ReadOnly="true"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblEmail" runat="server" Text="Email:"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtEmail" runat="server" ReadOnly="true"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblAddress" runat="server" Text="Address:"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtAddress" runat="server"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblContactNumber" runat="server" Text="Contact Number:"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtContactNumber" runat="server"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblGender" runat="server" Text="Gender:"></asp:Label>

</td>

<td>

<asp:DropDownList ID="ddlGender" runat="server">

<asp:ListItem Text="Select Gender" Value="" />

<asp:ListItem Text="Male" Value="Male" />

<asp:ListItem Text="Female" Value="Female" />

</asp:DropDownList>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblShopName" runat="server" Text="Shop Name:"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtShopName" runat="server" ReadOnly="true"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblImage" runat="server" Text="Image:"></asp:Label>

</td>

<td>

<asp:Image ID="imgUser" runat="server" />

<asp:FileUpload ID="fileUploadImage" runat="server" />

</td>

</tr>

</table>

<asp:Button ID="btnEdit" runat="server" Text="Edit Profile" OnClick="btnEdit\_Click" />

<asp:Button ID="btnUpdate" runat="server" Text="Update Profile" OnClick="btnUpdate\_Click" Visible="false" />

</div>

</asp:Content>

**:=> UserProfile.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using System.IO;

public partial class User\_UserProfile : System.Web.UI.Page

{

string connectionString = System.Configuration.ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

LoadUserProfile();

}

}

private void LoadUserProfile()

{

int userId = GetUserIdFromSession();

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT \* FROM [User] WHERE UserID = @UserID";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@UserID", userId);

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

txtUsername.Text = reader["UserName"].ToString();

txtEmail.Text = reader["Email"].ToString();

txtAddress.Text = reader["Address"].ToString();

txtContactNumber.Text = reader["ContactNumber"].ToString();

ddlGender.SelectedValue = reader["Gender"].ToString();

txtShopName.Text = reader["ShopName"].ToString();

string imagePath = reader["Image"].ToString();

if (!string.IsNullOrEmpty(imagePath))

{

imgUser.ImageUrl = imagePath;

}

}

reader.Close();

}

catch (Exception ex)

{

Response.Write("Error loading user profile: " + ex.Message);

}

}

}

}

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

{

EnableEditMode(true);

btnUpdate.Visible = true;

btnEdit.Visible = false;

fileUploadImage.Visible = true;

txtShopName.ReadOnly = false;

txtUsername.ReadOnly = false;

}

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

{

int userId = GetUserIdFromSession();

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query;

SqlCommand command;

if (fileUploadImage.HasFile)

{

query = "UPDATE [User] SET UserName = @UserName, Email = @Email, Address = @Address, ContactNumber = @ContactNumber, " +

"Gender = @Gender, ShopName = @ShopName, Image = @Image " +

"WHERE UserID = @UserID";

command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@Image", SaveImageAndGetPath(fileUploadImage));

}

else

{

query = "UPDATE [User] SET UserName = @UserName, Email = @Email, Address = @Address, ContactNumber = @ContactNumber, " +

"Gender = @Gender, ShopName = @ShopName " +

"WHERE UserID = @UserID";

command = new SqlCommand(query, connection);

}

command.Parameters.AddWithValue("@UserName", txtUsername.Text);

command.Parameters.AddWithValue("@Email", txtEmail.Text);

command.Parameters.AddWithValue("@Address", txtAddress.Text);

command.Parameters.AddWithValue("@ContactNumber", txtContactNumber.Text);

command.Parameters.AddWithValue("@Gender", ddlGender.SelectedValue);

command.Parameters.AddWithValue("@ShopName", txtShopName.Text);

command.Parameters.AddWithValue("@UserID", userId);

try

{

connection.Open();

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

EnableEditMode(false);

btnUpdate.Visible = false;

btnEdit.Visible = true;

LoadUserProfile();

}

else

{

Response.Write("Profile update failed.");

}

}

catch (Exception ex)

{

Response.Write("Error updating user profile: " + ex.Message);

}

}

}

private string SaveImageAndGetPath(FileUpload fileUpload)

{

string imageDirectory = Server.MapPath("~/User/Upload/");

string imagePath = "";

if (fileUpload.HasFile)

{

string fileName = Path.GetFileName(fileUpload.FileName);

imagePath = Path.Combine(imageDirectory, fileName);

fileUpload.SaveAs(imagePath);

imagePath = "~/User/Upload/" + fileName;

}

return imagePath;

}

private int GetUserIdFromSession()

{

if (Session["UserID"] != null)

{

return Convert.ToInt32(Session["UserID"]);

}

else

{

return -1;

}

}

private void EnableEditMode(bool enable)

{

txtUsername.Enabled = enable;

txtEmail.Enabled = false;

txtAddress.Enabled = enable;

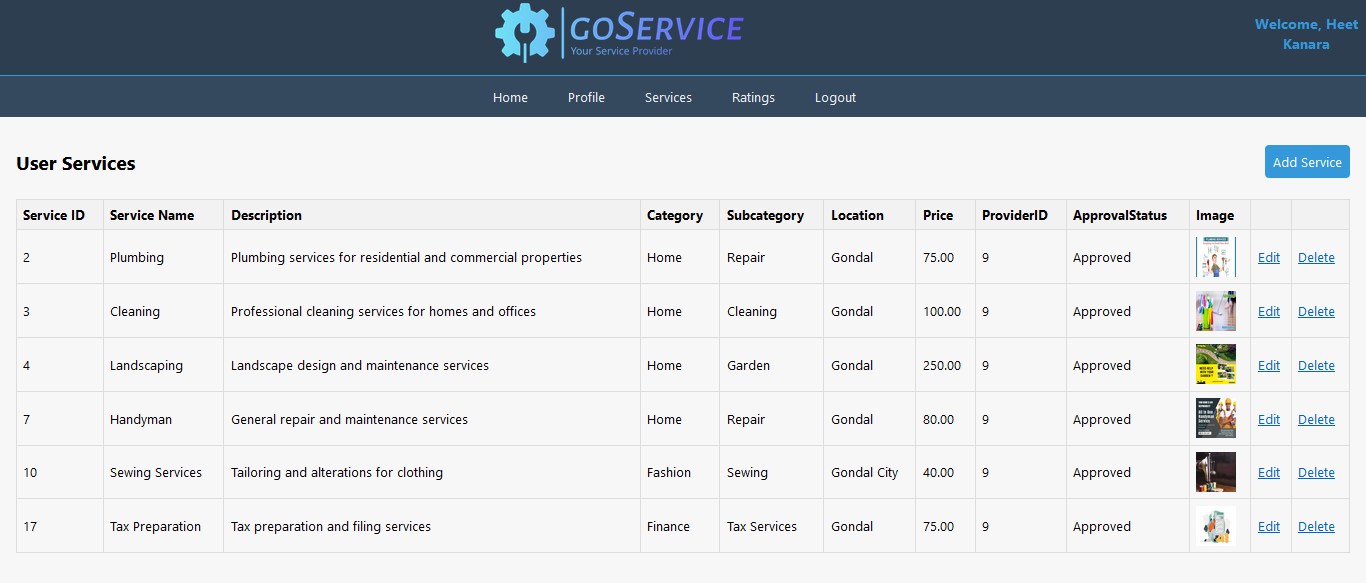
txtContactNumber.Enabled = enable;

ddlGender.Enabled = enable;

txtShopName.Enabled = enable;

}

}

**User Services**

**:=> UserServices.aspx**

<%@ Page Language="C#" MasterPageFile="~/User/UserMasterPage.master" AutoEventWireup="true" CodeFile="UserServices.aspx.cs" Inherits="User\_UserServices" Title="User Services" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div>

<h2>User Services

<asp:LinkButton ID="LinkButton1" runat="server">

<a href="UserAddService.aspx">Add Service</a>

</asp:LinkButton>

</h2>

<asp:GridView ID="gvServices" runat="server" AutoGenerateColumns="False" OnRowEditing="gvServices\_RowEditing"

OnRowUpdating="gvServices\_RowUpdating" OnRowCancelingEdit="gvServices\_RowCancelingEdit"

OnRowDeleting="gvServices\_RowDeleting" OnRowDataBound="gvServices\_RowDataBound" DataKeyNames="ServiceID" CssClass="table">

<Columns>

<asp:BoundField DataField="ServiceID" HeaderText="Service ID" ReadOnly="true" SortExpression="ServiceID" />

<asp:TemplateField HeaderText="Service Name">

<EditItemTemplate>

<asp:TextBox ID="txtServiceName" runat="server" Text='<%# Bind("ServiceName") %>'></asp:TextBox>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="lblServiceName" runat="server" Text='<%# Bind("ServiceName") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Description">

<EditItemTemplate>

<asp:TextBox ID="txtDescription" runat="server" Text='<%# Bind("Description") %>' TextMode="MultiLine" Rows="3"></asp:TextBox>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="lblDescription" runat="server" Text='<%# Bind("Description") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Category">

<EditItemTemplate>

<asp:TextBox ID="txtCategory" runat="server" Text='<%# Bind("Category") %>'></asp:TextBox>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="lblCategory" runat="server" Text='<%# Bind("Category") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Subcategory">

<EditItemTemplate>

<asp:TextBox ID="txtSubcategory" runat="server" Text='<%# Bind("Subcategory") %>'></asp:TextBox>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="lblSubcategory" runat="server" Text='<%# Bind("Subcategory") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Location">

<EditItemTemplate>

<asp:TextBox ID="txtLocation" runat="server" Text='<%# Bind("Location") %>'></asp:TextBox>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="lblLocation" runat="server" Text='<%# Bind("Location") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Price">

<EditItemTemplate>

<asp:TextBox ID="txtPrice" runat="server" Text='<%# Bind("Price") %>'></asp:TextBox>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="lblPrice" runat="server" Text='<%# Bind("Price") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="ProviderID">

<EditItemTemplate>

<asp:Label ID="lblProviderID" runat="server" Text='<%# Bind("ProviderID") %>'></asp:Label>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="lblProviderID" runat="server" Text='<%# Bind("ProviderID") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="ApprovalStatus">

<EditItemTemplate>

<asp:Label ID="lblApprovalStatus" runat="server" Text='<%# Bind("ApprovalStatus") %>'></asp:Label>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="lblApprovalStatus" runat="server" Text='<%# Bind("ApprovalStatus") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Image">

<EditItemTemplate>

<asp:FileUpload ID="fileUploadImage" runat="server" />

</EditItemTemplate>

<ItemTemplate>

<asp:Image ID="imgService" runat="server" Width="50px" Height="50px" ImageUrl='<%# Eval("ImagePath") %>' />

</ItemTemplate>

</asp:TemplateField>

<asp:CommandField ShowEditButton="True" />

<asp:CommandField ShowDeleteButton="True" />

</Columns>

</asp:GridView>

</div>

</asp:Content>

**:=> UserServices.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using System.IO;

public partial class User\_UserServices : System.Web.UI.Page

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

private FileUpload fileUploadImage;

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

{

if (!IsPostBack)

{

LoadServices(GetLoggedInUserId());

}

}

protected void LoadServices(int userId)

{

string query = "SELECT \* FROM Services WHERE ProviderID = @UserID";

DataTable dataTable = ExecuteQuery(query, new SqlParameter("@UserID", userId));

gvServices.DataSource = dataTable;

gvServices.DataBind();

}

private int GetLoggedInUserId()

{

if (Session["UserID"] != null)

{

int userId;

if (int.TryParse(Session["UserID"].ToString(), out userId))

{

return userId;

}

}

Response.Redirect("~/Login.aspx");

return -1;

}

protected void gvServices\_RowEditing(object sender, GridViewEditEventArgs e)

{

gvServices.EditIndex = e.NewEditIndex;

LoadServices(GetLoggedInUserId());

}

protected void gvServices\_RowUpdating(object sender, GridViewUpdateEventArgs e)

{

fileUploadImage = (FileUpload)gvServices.Rows[e.RowIndex].FindControl("fileUploadImage");

int serviceID = Convert.ToInt32(gvServices.DataKeys[e.RowIndex].Value);

TextBox txtServiceName = (TextBox)gvServices.Rows[e.RowIndex].FindControl("txtServiceName");

TextBox txtDescription = (TextBox)gvServices.Rows[e.RowIndex].FindControl("txtDescription");

TextBox txtCategory = (TextBox)gvServices.Rows[e.RowIndex].FindControl("txtCategory");

TextBox txtSubcategory = (TextBox)gvServices.Rows[e.RowIndex].FindControl("txtSubcategory");

TextBox txtLocation = (TextBox)gvServices.Rows[e.RowIndex].FindControl("txtLocation");

TextBox txtPrice = (TextBox)gvServices.Rows[e.RowIndex].FindControl("txtPrice");

if (txtServiceName != null && txtDescription != null && txtCategory != null &&

txtSubcategory != null && txtLocation != null && txtPrice != null)

{

string serviceName = txtServiceName.Text;

string description = txtDescription.Text;

string category = txtCategory.Text;

string subcategory = txtSubcategory.Text;

string location = txtLocation.Text;

decimal price = Convert.ToDecimal(txtPrice.Text);

UpdateService(serviceID, serviceName, description, category, subcategory, location, price, fileUploadImage);

gvServices.EditIndex = -1;

LoadServices(GetLoggedInUserId());

}

}

protected void gvServices\_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)

{

gvServices.EditIndex = -1;

LoadServices(GetLoggedInUserId());

}

protected void gvServices\_RowDeleting(object sender, GridViewDeleteEventArgs e)

{

int serviceID = Convert.ToInt32(gvServices.DataKeys[e.RowIndex].Value);

DeleteService(serviceID);

LoadServices(GetLoggedInUserId());

}

private void UpdateService(int serviceID, string serviceName, string description, string category, string subcategory, string location, decimal price, FileUpload fileUploadImage)

{

string query;

SqlParameter[] parameters;

if (fileUploadImage.HasFile)

{

query = "UPDATE Services SET ServiceName = @ServiceName, Description = @Description, " +

"Category = @Category, Subcategory = @Subcategory, Location = @Location, " +

"Price = @Price, ImagePath = @ImagePath " +

"WHERE ServiceID = @ServiceID";

parameters = new SqlParameter[]

{

new SqlParameter("@ServiceName", serviceName),

new SqlParameter("@Description", description),

new SqlParameter("@Category", category),

new SqlParameter("@Subcategory", subcategory),

new SqlParameter("@Location", location),

new SqlParameter("@Price", price),

new SqlParameter("@ServiceID", serviceID),

new SqlParameter("@ImagePath", SaveImageAndGetPath(fileUploadImage))

};

}

else

{

query = "UPDATE Services SET ServiceName = @ServiceName, Description = @Description, " +

"Category = @Category, Subcategory = @Subcategory, Location = @Location, " +

"Price = @Price " +

"WHERE ServiceID = @ServiceID";

parameters = new SqlParameter[]

{

new SqlParameter("@ServiceName", serviceName),

new SqlParameter("@Description", description),

new SqlParameter("@Category", category),

new SqlParameter("@Subcategory", subcategory),

new SqlParameter("@Location", location),

new SqlParameter("@Price", price),

new SqlParameter("@ServiceID", serviceID)

};

}

ExecuteNonQuery(query, parameters);

}

private string SaveImageAndGetPath(FileUpload fileUpload)

{

string imageDirectory = Server.MapPath("Upload/");

string imagePath = "";

if (fileUpload.HasFile)

{

string fileName = Path.GetFileName(fileUpload.FileName);

imagePath = Path.Combine(imageDirectory, fileName);

fileUpload.SaveAs(imagePath);

imagePath = "~/User/Upload/" + fileName;

}

return imagePath;

}

private void DeleteService(int serviceID)

{

string query = "DELETE FROM Services WHERE ServiceID = @ServiceID";

ExecuteNonQuery(query, new SqlParameter("@ServiceID", serviceID));

}

private DataTable ExecuteQuery(string query, params SqlParameter[] parameters)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

using (SqlDataAdapter adapter = new SqlDataAdapter(query, connection))

{

adapter.SelectCommand.Parameters.AddRange(parameters);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

return dataTable;

}

}

}

private void ExecuteNonQuery(string query, params SqlParameter[] parameters)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddRange(parameters);

connection.Open();

command.ExecuteNonQuery();

}

}

}

protected void gvServices\_RowDataBound(object sender, GridViewRowEventArgs e)

{

if (e.Row.RowType == DataControlRowType.DataRow)

{

FileUpload fileUploadImage = (FileUpload)e.Row.FindControl("fileUploadImage");

Image imgService = (Image)e.Row.FindControl("imgService");

if (fileUploadImage != null && imgService != null)

{

if (gvServices.EditIndex == e.Row.RowIndex)

{

fileUploadImage.Visible = true;

imgService.Visible = false;

}

else

{

fileUploadImage.Visible = false;

imgService.Visible = true;

}

}

}

}

}

**User Add Service**



**:=> UserAddService.aspx**

<%@ Page Language="C#" MasterPageFile="~/User/UserMasterPage.master" AutoEventWireup="true" CodeFile="UserAddService.aspx.cs" Inherits="User\_UserAddService" Title="Add Service" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

.form-container {

background-color: #f8f9fa;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 60%;

box-sizing: border-box;

text-align: left;

margin: auto;

}

h2 {

color: #333;

}

label {

display: block;

margin-bottom: 5px;

color: #555;

}

.form-container input[type="text"],

.form-container textarea,

.form-container select {

width: calc(100% - 16px);

padding: 8px;

margin-bottom: 10px;

box-sizing: border-box;

border: 1px solid #ced4da;

border-radius: 4px;

display: inline-block;

}

.form-container input[type="text"],

.form-container textarea {

display: block;

}

.validation-summary-errors {

color: #ff0000;

margin-bottom: 10px;

}

.button-container {

text-align: center;

margin-top: 15px;

}

.btnAddService {

background-color: #007bff;

color: white;

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

}

.btnAddService:hover {

background-color: #0056b3;

}

.file-upload-container {

margin-top: 15px;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div class="form-container">

<h2>Add New Service</h2>

<asp:Label ID="lblServiceName" runat="server" Text="Service Name:"></asp:Label>

<asp:TextBox ID="txtServiceName" runat="server"></asp:TextBox>

<asp:RequiredFieldValidator ID="rfvServiceName" runat="server"

ControlToValidate="txtServiceName"

ErrorMessage="Please enter a service name."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<br />

<asp:Label ID="lblDescription" runat="server" Text="Description:"></asp:Label>

<asp:TextBox ID="txtDescription" runat="server" TextMode="MultiLine"></asp:TextBox>

<asp:RequiredFieldValidator ID="rfvDescription" runat="server"

ControlToValidate="txtDescription"

ErrorMessage="Please enter a description."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<br />

<asp:Label ID="Label1" runat="server" Text="Category:"></asp:Label>

<asp:DropDownList ID="ddlCategory" runat="server" OnSelectedIndexChanged="ddlCategory\_SelectedIndexChanged" AutoPostBack="true">

<asp:ListItem Text="Select Existing Category" Value="" />

<asp:ListItem Text="Add New Category" Value="AddNew" />

</asp:DropDownList>

<asp:TextBox ID="TextBox1" runat="server" placeholder="New Category" Enabled="false"></asp:TextBox>

<asp:RequiredFieldValidator ID="rfvCategory" runat="server"

ControlToValidate="ddlCategory"

ErrorMessage="Please select an existing category or enter a new one."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<asp:RequiredFieldValidator ID="rfvNewCategory" runat="server"

ControlToValidate="TextBox1"

InitialValue=""

ErrorMessage="Please enter a new category."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<br />

<asp:Label ID="Label2" runat="server" Text="Subcategory:"></asp:Label>

<asp:DropDownList ID="ddlSubcategory" runat="server" OnSelectedIndexChanged="ddlSubcategory\_SelectedIndexChanged" AutoPostBack="true">

<asp:ListItem Text="Select Existing Subcategory" Value="" />

<asp:ListItem Text="Add New Subcategory" Value="AddNew" />

</asp:DropDownList>

<asp:TextBox ID="txtSubcategory" runat="server" placeholder="New Subcategory" Enabled="false"></asp:TextBox>

<asp:RequiredFieldValidator ID="rfvSubcategory" runat="server"

ControlToValidate="ddlSubcategory"

ErrorMessage="Please select a subcategory or enter a new one."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<asp:RequiredFieldValidator ID="rfvTxtSubcategory" runat="server"

ControlToValidate="txtSubcategory"

ErrorMessage="Please enter a new subcategory."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<asp:RegularExpressionValidator ID="revTxtSubcategory" runat="server"

ControlToValidate="txtSubcategory"

ErrorMessage="Invalid subcategory format."

ValidationExpression="^[a-zA-Z0-9\s]+$"

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RegularExpressionValidator>

<br />

<asp:Label ID="Label3" runat="server" Text="Location:"></asp:Label>

<asp:DropDownList ID="ddlLocation" runat="server" OnSelectedIndexChanged="ddlLocation\_SelectedIndexChanged" AutoPostBack="true">

<asp:ListItem Text="Select Existing Location" Value="" />

<asp:ListItem Text="Add New Location" Value="AddNew" />

</asp:DropDownList>

<asp:TextBox ID="TextBox3" runat="server" placeholder="New Location" Enabled="false"></asp:TextBox>

<asp:RequiredFieldValidator ID="rfvLocation" runat="server"

ControlToValidate="ddlLocation"

ErrorMessage="Please select an existing location or enter a new one."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<asp:RequiredFieldValidator ID="rfvNewLocation" runat="server"

ControlToValidate="TextBox3"

ErrorMessage="Please enter a new location."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<br />

<asp:Label ID="lblPrice" runat="server" Text="Price:"></asp:Label>

<asp:TextBox ID="txtPrice" runat="server"></asp:TextBox>

<asp:RequiredFieldValidator ID="rfvPrice" runat="server"

ControlToValidate="txtPrice"

ErrorMessage="Please enter a price."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<br />

<asp:Label ID="lblImage" runat="server" Text="Service Image:"></asp:Label>

<asp:FileUpload ID="FileUploadService" runat="server" />

<asp:RequiredFieldValidator ID="rfvFileUploadService" runat="server"

ControlToValidate="FileUploadService"

ErrorMessage="Please select a service image."

Display="Dynamic"

SetFocusOnError="true"

EnableClientScript="true"

ValidationGroup="AddServiceValidation">

</asp:RequiredFieldValidator>

<br />

<asp:Button ID="btnAddService" runat="server" Text="Add Service" OnClick="btnAddService\_Click" ValidationGroup="AddServiceValidation" />

</div>

</asp:Content>

**:=> UserAddService.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using System.IO;

public partial class User\_UserAddService : System.Web.UI.Page

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

LoadCategories();

LoadSubcategories(ddlCategory.SelectedValue);

LoadLocations(ddlSubcategory.SelectedValue);

}

}

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

{

int providerID = GetProviderIDFromSession();

string serviceName = txtServiceName.Text;

string description = txtDescription.Text;

string category = ddlCategory.SelectedValue == "AddNew" ? TextBox1.Text : ddlCategory.SelectedValue;

string subcategory = ddlSubcategory.SelectedValue == "AddNew" ? txtSubcategory.Text : ddlSubcategory.SelectedValue;

string location = ddlLocation.SelectedValue == "AddNew" ? TextBox3.Text : ddlLocation.SelectedValue;

decimal price = Convert.ToDecimal(txtPrice.Text);

string approvalStatus = "Pending";

string filePath = Server.MapPath("Upload/");

string fileName = System.IO.Path.GetFileName(FileUploadService.FileName);

string extension = System.IO.Path.GetExtension(fileName);

HttpPostedFile postedFile = FileUploadService.PostedFile;

int size = postedFile.ContentLength;

if (FileUploadService.HasFile && IsImageFile(extension) && size <= 5000000)

{

string imagePath = "~/User/Upload/" + fileName;

FileUploadService.SaveAs(System.IO.Path.Combine(filePath, fileName));

AddServiceWithImage(providerID, serviceName, description, category, subcategory, location, price, approvalStatus, imagePath);

Response.Redirect("UserServices.aspx");

}

else

{

}

}

private void LoadCategories()

{ using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT DISTINCT Category FROM Services";

SqlCommand cmd = new SqlCommand(query, connection);

try

{

connection.Open();

SqlDataReader reader = cmd.ExecuteReader();

ddlCategory.Items.Clear();

ddlCategory.Items.Add(new ListItem("Select Existing Category", string.Empty));

ddlCategory.Items.Add(new ListItem("Add New Category", "AddNew"));

while (reader.Read())

{

ddlCategory.Items.Add(reader["Category"].ToString());

}

reader.Close();

}

catch (Exception ex)

{

Console.WriteLine("An error occurred while loading categories: " + ex.Message);

}

}

}

private void LoadSubcategories(string selectedCategory)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT DISTINCT Subcategory FROM Services WHERE Category = @SelectedCategory";

SqlCommand cmd = new SqlCommand(query, connection);

cmd.Parameters.AddWithValue("@SelectedCategory", selectedCategory);

try

{

connection.Open();

SqlDataReader reader = cmd.ExecuteReader();

ddlSubcategory.Items.Clear();

ddlSubcategory.Items.Add(new ListItem("Select Existing Subcategory", string.Empty));

ddlSubcategory.Items.Add(new ListItem("Add New Subcategory", "AddNew"));

while (reader.Read())

{

ddlSubcategory.Items.Add(reader["Subcategory"].ToString());

}

reader.Close();

}

catch (Exception ex)

{

Console.WriteLine("An error occurred while loading subcategories: " + ex.Message);

}

}

}

private void LoadLocations(string selectedSubcategory)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT DISTINCT Location FROM Services WHERE Subcategory = @SelectedSubcategory";

SqlCommand cmd = new SqlCommand(query, connection);

cmd.Parameters.AddWithValue("@SelectedSubcategory", selectedSubcategory);

try

{

connection.Open();

SqlDataReader reader = cmd.ExecuteReader();

ddlLocation.Items.Clear();

ddlLocation.Items.Add(new ListItem("Select Existing Location", string.Empty));

ddlLocation.Items.Add(new ListItem("Add New Location", "AddNew"));

while (reader.Read())

{

ddlLocation.Items.Add(reader["Location"].ToString());

}

reader.Close();

}

catch (Exception ex)

{

Console.WriteLine("An error occurred while loading locations: " + ex.Message);

}

}

}

private int GetProviderIDFromSession()

{

return Session["UserID"] != null ? Convert.ToInt32(Session["UserID"]) : -1;

}

private void AddServiceWithImage(int providerID, string serviceName, string description, string category, string subcategory, string location, decimal price, string approvalStatus, string imagePath)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "INSERT INTO Services (ProviderID, ServiceName, Description, Category, Subcategory, Location, Price, ApprovalStatus, ImagePath) " +

"VALUES (@ProviderID, @ServiceName, @Description, @Category, @Subcategory, @Location, @Price, @ApprovalStatus, @ImagePath)";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@ProviderID", providerID);

command.Parameters.AddWithValue("@ServiceName", serviceName);

command.Parameters.AddWithValue("@Description", description);

command.Parameters.AddWithValue("@Category", category);

command.Parameters.AddWithValue("@Subcategory", subcategory);

command.Parameters.AddWithValue("@Location", location);

command.Parameters.AddWithValue("@Price", price);

command.Parameters.AddWithValue("@ApprovalStatus", approvalStatus);

command.Parameters.AddWithValue("@ImagePath", imagePath);

connection.Open();

command.ExecuteNonQuery();

}

}

}

private bool IsImageFile(string extension)

{

string[] allowedExtensions = { ".jpg", ".jpeg", ".png" };

return Array.IndexOf(allowedExtensions, extension.ToLower()) != -1;

}

protected void ddlCategory\_SelectedIndexChanged(object sender, EventArgs e)

{

LoadSubcategories(ddlCategory.SelectedValue);

LoadLocations(ddlSubcategory.SelectedValue);

UpdateDropdownState(ddlCategory, TextBox1, rfvCategory, rfvNewCategory);

}

protected void ddlSubcategory\_SelectedIndexChanged(object sender, EventArgs e)

{

LoadLocations(ddlSubcategory.SelectedValue);

UpdateDropdownState(ddlSubcategory, txtSubcategory, rfvSubcategory, rfvTxtSubcategory);

}

protected void ddlLocation\_SelectedIndexChanged(object sender, EventArgs e)

{

UpdateDropdownState(ddlLocation, TextBox3, rfvLocation, rfvNewLocation);

}

private void UpdateDropdownState(DropDownList dropdown, TextBox textBox, RequiredFieldValidator dropdownValidator, RequiredFieldValidator newValidator)

{

if (dropdown.SelectedValue == "AddNew")

{

textBox.Enabled = true;

dropdownValidator.Enabled = false;

newValidator.Enabled = true;

}

else

{

textBox.Enabled = false;

dropdownValidator.Enabled = true;

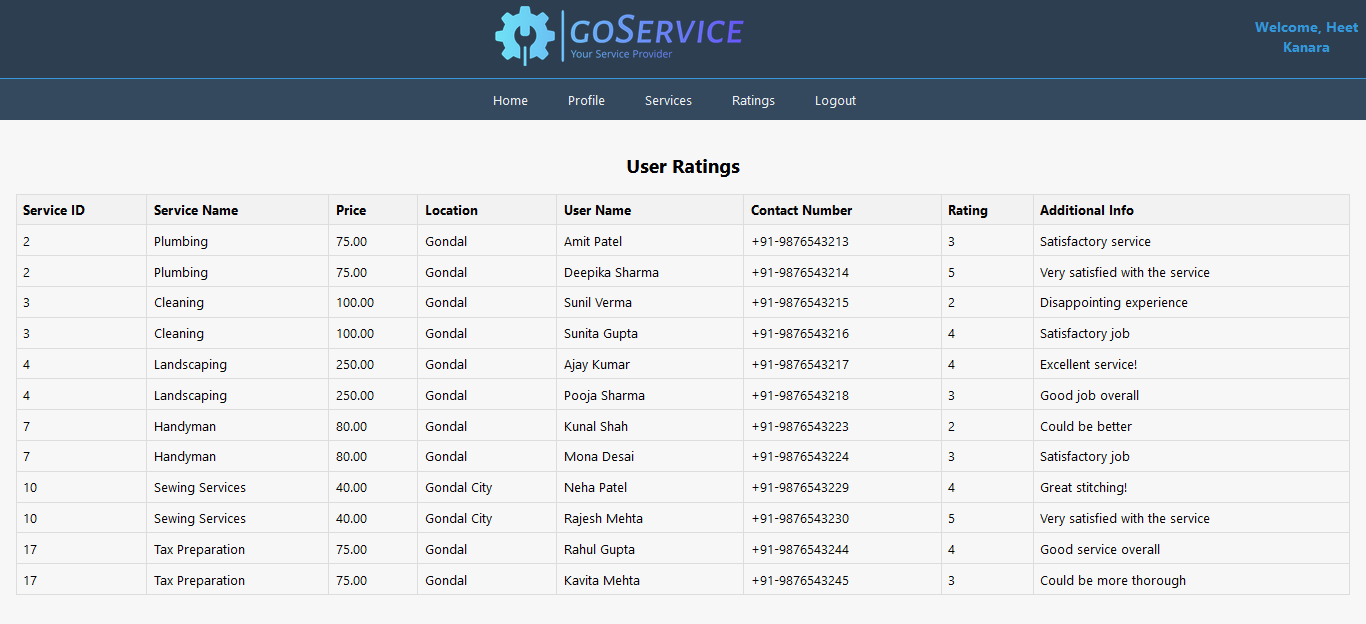
newValidator.Enabled = false;

}

}

}

**User Ratings**



**:=> UserRating.aspx**

<%@ Page Language="C#" MasterPageFile="~/User/UserMasterPage.master" AutoEventWireup="true" CodeFile="UserRating.aspx.cs" Inherits="User\_UserRating" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style type="text/css">

table {

border-collapse: collapse;

width: 100%;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<h2>User Ratings</h2>

<asp:GridView ID="GridViewRatings" runat="server" AutoGenerateColumns="False" CssClass="gridview">

<Columns>

<asp:BoundField DataField="ServiceID" HeaderText="Service ID" SortExpression="ServiceID" />

<asp:BoundField DataField="ServiceName" HeaderText="Service Name" SortExpression="ServiceName" />

<asp:BoundField DataField="Price" HeaderText="Price" SortExpression="Price" />

<asp:BoundField DataField="Location" HeaderText="Location" SortExpression="Location" />

<asp:BoundField DataField="UserName" HeaderText="User Name" SortExpression="UserName" />

<asp:BoundField DataField="ContactNumber" HeaderText="Contact Number" SortExpression="ContactNumber" />

<asp:BoundField DataField="RatingValue" HeaderText="Rating" SortExpression="RatingValue" />

<asp:BoundField DataField="AdditionalInfo" HeaderText="Additional Info" SortExpression="AdditionalInfo" />

</Columns>

</asp:GridView>

</asp:Content>

**:=> UserRating.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public partial class User\_UserRating : System.Web.UI.Page

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

if (User.Identity.IsAuthenticated)

{

string userId = Session["UserID"].ToString();

LoadUserRatings(userId);

}

else

{

Response.Redirect("~/Account/Login.aspx");

}

}

}

private void LoadUserRatings(string userId)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT r.ServiceID, s.ServiceName, s.Price, s.Location, r.UserName, r.ContactNumber, r.RatingValue, r.AdditionalInfo " +

"FROM Rating r " +

"INNER JOIN Services s ON r.ServiceID = s.ServiceID " +

"WHERE r.ProviderID = @UserID";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@UserID", userId);

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader();

GridViewRatings.DataSource = reader;

GridViewRatings.DataBind();

reader.Close();

}

catch (Exception ex)

{

Response.Write("Error loading user ratings: " + ex.Message);

}

}

}

}

}

**User Logout**

**:=> UserLogout.aspx**

<%@ Page Language="C#" MasterPageFile="~/User/UserMasterPage.master" AutoEventWireup="true" CodeFile="UserLogout.aspx.cs" Inherits="User\_UserLogout" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

</asp:Content>

**:=> UserLogout.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

public partial class User\_UserLogout : System.Web.UI.Page

{

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

{

Session.Clear();

Session.Abandon();

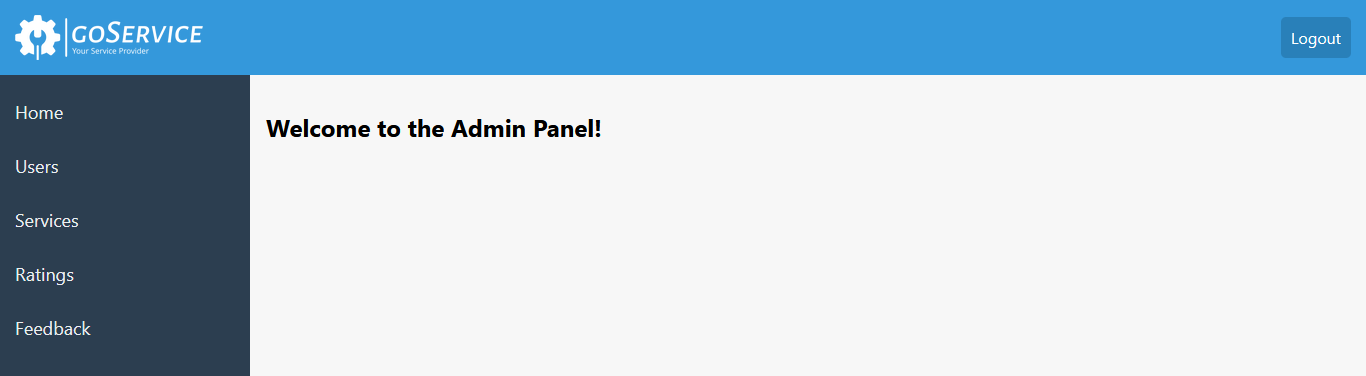
Response.Redirect("../Login.aspx");

}

}

**ADMIN SIDE**

**Admin Home Page**



**:=> AdminHome.aspx**

<%@ Page Language="C#" MasterPageFile="~/Admin/AdminMasterPage.master" AutoEventWireup="true" CodeFile="AdminHome.aspx.cs" Inherits="Admin\_AdminHome" Title="Admin Home" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div>

<h2>Welcome to the Admin Panel!</h2>

</div>

</asp:Content>

**:=> AdminHome.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

public partial class Admin\_AdminHome : System.Web.UI.Page

{

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

{

}

}

**:=> AdminMasterPage.aspx**

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="AdminMasterPage.master.cs" Inherits="Admin\_AdminMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head id="Head1" runat="server">

<title>Admin Panel</title>

<style type="text/css">

body {

margin: 0;

padding: 0;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background-color: #f7f7f7;

}

.header {

position: fixed;

top: 0;

left: 0;

right: 0;

z-index: 999;

background-color: #3498db;

padding: 15px;

display: flex;

justify-content: space-between;

align-items: center;

color: white;

}

.logo img {

height: 45px;

margin-right: 10px;

}

.logout {

color: white;

text-decoration: none;

padding: 10px;

background-color: #2980b9;

border-radius: 5px;

}

.logout:hover {

background-color: #1d5b87;

}

.sidebar {

position: fixed;

top: 60px;

bottom: 0;

width: 250px;

background-color: #2c3e50;

padding-top: 10px;

overflow-y: auto;

}

.sidebar a {

padding: 15px;

text-decoration: none;

font-size: 18px;

color: #ecf0f1;

display: block;

transition: 0.3s;

}

.sidebar a:hover {

color: #3498db;

}

.content {

margin-left: 250px;

padding: 16px;

padding-top: 80px;

}

</style>

<asp:ContentPlaceHolder id="head" runat="server">

</asp:ContentPlaceHolder>

</head>

<body>

<form id="form1" runat="server">

<div class="header">

<div class="logo">

<img src="../images/goservice-high-resolution-logo-white-transparent.png" />

</div>

<a href="AdminLogout.aspx" class="logout">Logout</a>

</div>

<div class="sidebar">

<a href="AdminHome.aspx">Home</a>

<a href="AdminUsers.aspx">Users</a>

<a href="AdminServices.aspx">Services</a>

<a href="AdminRatings.aspx">Ratings</a>

<a href="AdminFeedbacks.aspx">Feedback</a>

</div>

<div class="content">

<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

</asp:ContentPlaceHolder>

</div>

</form>

</body>

</html>

**:=> AdminMasterPage.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Xml.Linq;

public partial class Admin\_AdminMasterPage : System.Web.UI.MasterPage

{

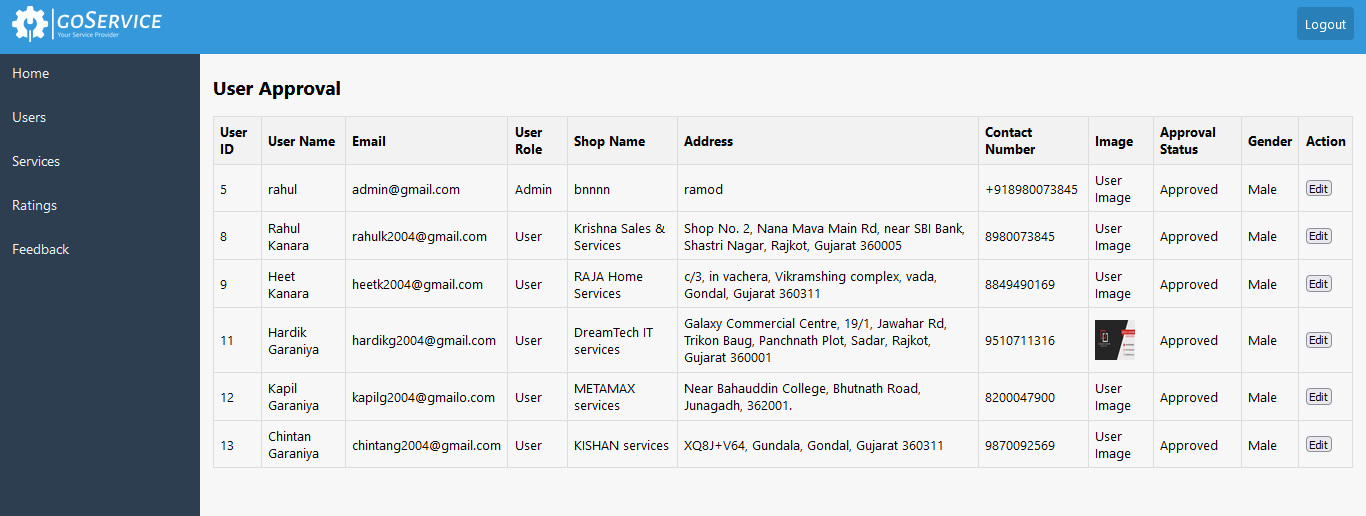
protected void Page\_Load(object sender, EventArgs e)

{

}

}

**User Approval**



**:=> AdminUsers.aspx**

<%@ Page Language="C#" MasterPageFile="~/Admin/AdminMasterPage.master" AutoEventWireup="true" CodeFile="AdminUsers.aspx.cs" Inherits="Admin\_AdminUsers" Title="Admin Users" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

.submit-button {

float: right;

margin: 10px;

cursor: pointer;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<h2>Welcome to the Admin Panel!</h2>

<div style="overflow-x: auto;">

<asp:GridView ID="GridViewUsers" runat="server" AutoGenerateColumns="False" DataKeyNames="UserID"

OnRowEditing="GridViewUsers\_RowEditing" OnRowCancelingEdit="GridViewUsers\_RowCancelingEdit"

OnRowUpdating="GridViewUsers\_RowUpdating" OnRowDataBound="GridViewUsers\_RowDataBound" EditIndex="-1">

<Columns>

<asp:BoundField DataField="UserID" HeaderText="User ID" ReadOnly="true" SortExpression="UserID" />

<asp:BoundField DataField="UserName" HeaderText="User Name" SortExpression="UserName" />

<asp:BoundField DataField="Email" HeaderText="Email" SortExpression="Email" />

<asp:BoundField DataField="UserRole" HeaderText="User Role" SortExpression="UserRole" />

<asp:BoundField DataField="ShopName" HeaderText="Shop Name" SortExpression="ShopName" />

<asp:BoundField DataField="Address" HeaderText="Address" SortExpression="Address" /> <asp:BoundField DataField="ContactNumber" HeaderText="Contact Number" SortExpression="ContactNumber" />

<asp:TemplateField HeaderText="Image">

<ItemTemplate>

<asp:Image ID="imgUser" runat="server" Width="50px" Height="50px" ImageUrl='<%# ResolveUrl(Eval("Image").ToString()) %>' AlternateText="User Image" />

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Approval Status">

<ItemTemplate>

<asp:Label ID="lblApprovalStatus" runat="server" Text='<%# Eval("Approval") %>'></asp:Label>

</ItemTemplate>

<EditItemTemplate>

<asp:RadioButton ID="rbApproved" runat="server" Text="Approve" GroupName='<%# Eval("UserID") %>' Checked='<%# Convert.ToBoolean(Eval("Approval")) %>' />

<asp:RadioButton ID="rbRejected" runat="server" Text="Reject" GroupName='<%# Eval("UserID") %>' Checked='<%# !Convert.ToBoolean(Eval("Approval")) %>' />

</EditItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="Gender" HeaderText="Gender" SortExpression="Gender" />

<asp:TemplateField HeaderText="Action">

<ItemTemplate>

<asp:Button ID="btnEdit" runat="server" Text="Edit" CommandName="Edit" />

</ItemTemplate>

<EditItemTemplate>

<asp:Button ID="btnUpdate" runat="server" Text="Update" CommandName="Update" />

<asp:Button ID="btnCancel" runat="server" Text="Cancel" CommandName="Cancel" />

</EditItemTemplate>

</asp:TemplateField>

</Columns>

</asp:GridView>

</div>

<asp:Label ID="lblSubmissionStatus" runat="server" Visible="false"></asp:Label>

</asp:Content>

**:=> AdminUsers.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public partial class Admin\_AdminUsers : System.Web.UI.Page

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

LoadUsers();

}

}

protected void LoadUsers()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT \* FROM [User] ORDER BY CASE WHEN ApprovalStatus = 'Pending' THEN 1 WHEN ApprovalStatus = 'Approved' THEN 2 ELSE 3 END";

SqlCommand cmd = new SqlCommand(query, connection);

try

{

connection.Open();

GridViewUsers.DataSource = cmd.ExecuteReader();

GridViewUsers.DataBind();

}

catch (Exception ex)

{

Console.WriteLine("An error occurred: " + ex.Message);

}

}

}

protected void GridViewUsers\_RowEditing(object sender, GridViewEditEventArgs e)

{

GridViewUsers.EditIndex = e.NewEditIndex;

LoadUsers();

}

protected void GridViewUsers\_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)

{

GridViewUsers.EditIndex = -1;

LoadUsers();

}

protected void GridViewUsers\_RowUpdating(object sender, GridViewUpdateEventArgs e)

{

string userID = GridViewUsers.DataKeys[e.RowIndex].Value.ToString();

RadioButton rbApproved = (RadioButton)GridViewUsers.Rows[e.RowIndex].FindControl("rbApproved");

RadioButton rbRejected = (RadioButton)GridViewUsers.Rows[e.RowIndex].FindControl("rbRejected");

bool approvalStatus = rbApproved.Checked;

UpdateUserApprovalStatus(userID, approvalStatus);

GridViewUsers.EditIndex = -1;

LoadUsers();

lblSubmissionStatus.Text = "Update successful";

lblSubmissionStatus.Visible = true;

}

private void UpdateUserApprovalStatus(string userID, bool approvalStatus)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string statusString = approvalStatus ? "Approved" : "Rejected";

string query = "UPDATE [User] SET ApprovalStatus = @ApprovalStatus WHERE UserID = @UserID";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.Add("@ApprovalStatus", SqlDbType.VarChar).Value = statusString;

command.Parameters.AddWithValue("@UserID", userID);

try

{

connection.Open();

command.ExecuteNonQuery();

}

catch (Exception ex)

{

Console.WriteLine("An error occurred: " + ex.Message);

}

}

}

protected void GridViewUsers\_RowDataBound(object sender, GridViewRowEventArgs e)

{

if (e.Row.RowType == DataControlRowType.DataRow && GridViewUsers.EditIndex == e.Row.RowIndex)

{

foreach (TableCell cell in e.Row.Cells)

{

foreach (Control control in cell.Controls)

{

TextBox textBox = control as TextBox;

if (textBox != null)

{

textBox.Enabled = false;

}

}

}

RadioButton rbApproved = (RadioButton)e.Row.FindControl("rbApproved");

RadioButton rbRejected = (RadioButton)e.Row.FindControl("rbRejected");

Label lblApprovalStatus = (Label)e.Row.FindControl("lblApprovalStatus");

rbApproved.Enabled = true;

rbRejected.Enabled = true;

if (Convert.ToBoolean(lblApprovalStatus.Text))

{

rbApproved.Checked = true;

}

else

{

rbRejected.Checked = true;

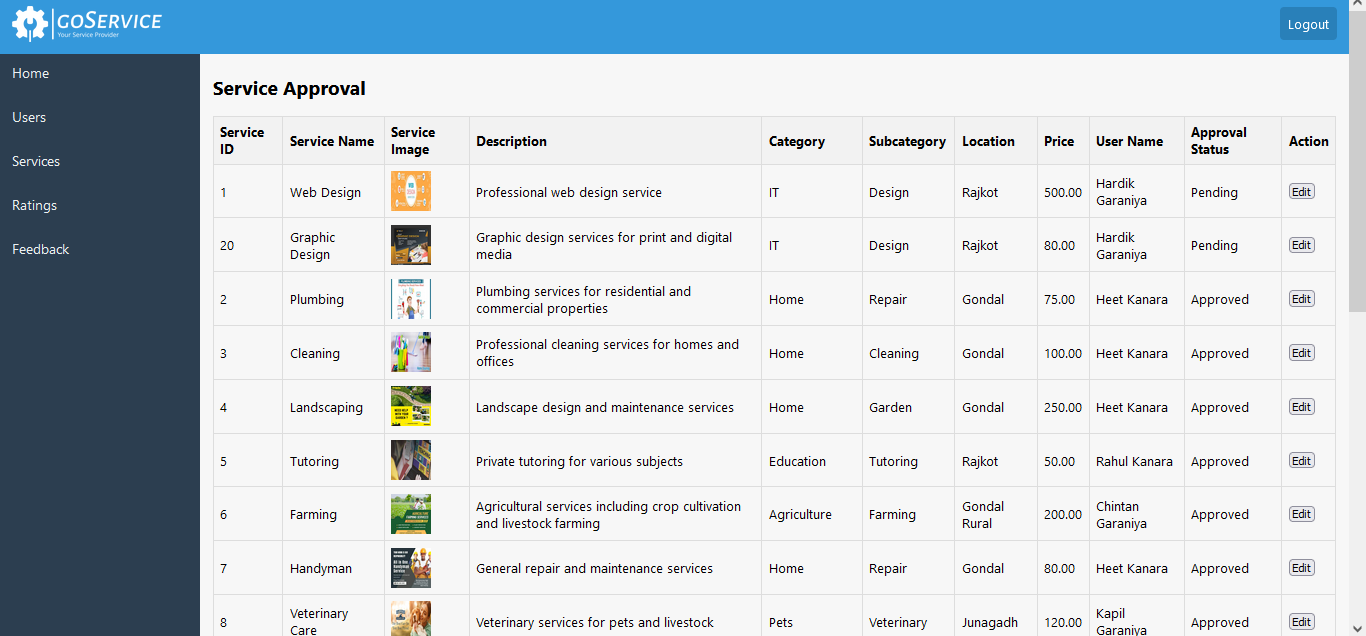
}

}

}

}

**Service Approval**



**:=> AdminServices.aspx**

<%@ Page Language="C#" MasterPageFile="~/Admin/AdminMasterPage.master" AutoEventWireup="true" CodeFile="AdminServices.aspx.cs" Inherits="Admin\_AdminServices" Title="Admin Services" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

.submit-button {

float: right;

margin: 10px;

cursor: pointer;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<h2>Welcome to the Admin Panel!</h2>

<div style="overflow-x: auto;">

<asp:GridView ID="GridViewServices" runat="server" AutoGenerateColumns="False" DataKeyNames="ServiceID"

OnRowEditing="GridViewServices\_RowEditing" OnRowCancelingEdit="GridViewServices\_RowCancelingEdit"

OnRowUpdating="GridViewServices\_RowUpdating" OnRowDataBound="GridViewServices\_RowDataBound" EditIndex="-1">

<Columns>

<asp:BoundField DataField="ServiceID" HeaderText="Service ID" ReadOnly="true" SortExpression="ServiceID" />

<asp:BoundField DataField="ServiceName" HeaderText="Service Name" SortExpression="ServiceName" />

<asp:TemplateField HeaderText="Service Image">

<ItemTemplate>

<asp:Image ID="imgService" runat="server" Width="50px" Height="50px" ImageUrl='<%# ResolveUrl(Eval("ImagePath").ToString()) %>' AlternateText="Service Image" />

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="Description" HeaderText="Description" SortExpression="Description" />

<asp:BoundField DataField="Category" HeaderText="Category" SortExpression="Category" />

<asp:BoundField DataField="Subcategory" HeaderText="Subcategory" SortExpression="Subcategory" />

<asp:BoundField DataField="Location" HeaderText="Location" SortExpression="Location" />

<asp:BoundField DataField="Price" HeaderText="Price" SortExpression="Price" />

<asp:BoundField DataField="UserName" HeaderText="User Name" SortExpression="UserName" />

<asp:TemplateField HeaderText="Approval Status">

<ItemTemplate>

<asp:Label ID="lblApprovalStatus" runat="server" Text='<%# Eval("ApprovalStatus") %>'></asp:Label>

</ItemTemplate>

<EditItemTemplate>

<asp:RadioButton ID="rbApproved" runat="server" Text="Approve" GroupName="ApprovalStatusGroup" Checked='<%# Eval("ApprovalStatus").ToString() == "Approved" %>' />

<asp:RadioButton ID="rbRejected" runat="server" Text="Reject" GroupName="ApprovalStatusGroup" Checked='<%# Eval("ApprovalStatus").ToString() == "Rejected" %>' />

</EditItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Action">

<ItemTemplate>

<asp:Button ID="btnEdit" runat="server" Text="Edit" CommandName="Edit" />

</ItemTemplate>

<EditItemTemplate>

<asp:Button ID="btnUpdate" runat="server" Text="Update" CommandName="Update" />

<asp:Button ID="btnCancel" runat="server" Text="Cancel" CommandName="Cancel" />

</EditItemTemplate>

</asp:TemplateField>

</Columns>

</asp:GridView>

</div>

<asp:Label ID="lblSubmissionStatus" runat="server" Visible="false"></asp:Label>

</asp:Content>

**:=> AdminServices.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public partial class Admin\_AdminServices : System.Web.UI.Page

{

private string connectionString = System.Configuration.ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

LoadServices();

}

}

protected void LoadServices()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT s.ServiceID, s.ServiceName, s.Description, s.Category, s.Subcategory, s.Location, s.Price, u.UserName, s.ApprovalStatus, s.ImagePath FROM Services s INNER JOIN [User] u ON s.ProviderID = u.UserID ORDER BY CASE WHEN s.ApprovalStatus = 'Pending' THEN 1 WHEN s.ApprovalStatus = 'Approved' THEN 2 ELSE 3 END";

SqlCommand cmd = new SqlCommand(query, connection);

try

{

connection.Open();

GridViewServices.DataSource = cmd.ExecuteReader();

GridViewServices.DataBind();

}

catch (Exception ex)

{

Console.WriteLine("An error occurred: " + ex.Message);

}

}

}

protected void GridViewServices\_RowEditing(object sender, GridViewEditEventArgs e)

{

GridViewServices.EditIndex = e.NewEditIndex;

LoadServices();

}

protected void GridViewServices\_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)

{

GridViewServices.EditIndex = -1;

LoadServices();

}

protected void GridViewServices\_RowUpdating(object sender, GridViewUpdateEventArgs e)

{

string serviceID = GridViewServices.DataKeys[e.RowIndex].Value.ToString();

RadioButton rbApproved = (RadioButton)GridViewServices.Rows[e.RowIndex].FindControl("rbApproved");

RadioButton rbRejected = (RadioButton)GridViewServices.Rows[e.RowIndex].FindControl("rbRejected");

string approvalStatus = rbApproved.Checked ? "Approved" : "Rejected";

UpdateApprovalStatus(serviceID, approvalStatus);

GridViewServices.EditIndex = -1;

LoadServices();

lblSubmissionStatus.Text = "Update successful";

lblSubmissionStatus.Visible = true;

}

private void UpdateApprovalStatus(string serviceID, string approvalStatus)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "UPDATE Services SET ApprovalStatus = @ApprovalStatus WHERE ServiceID = @ServiceID";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@ApprovalStatus", approvalStatus);

command.Parameters.AddWithValue("@ServiceID", serviceID);

try

{

connection.Open();

command.ExecuteNonQuery();

}

catch (Exception ex)

{

Console.WriteLine("An error occurred: " + ex.Message);

}

}

}

protected void GridViewServices\_RowDataBound(object sender, GridViewRowEventArgs e)

{

if (e.Row.RowType == DataControlRowType.DataRow && (e.Row.RowState & DataControlRowState.Edit) > 0)

{

foreach (TableCell cell in e.Row.Cells)

{

foreach (Control control in cell.Controls)

{

TextBox textBox = control as TextBox;

if (textBox != null)

{

textBox.Enabled = false;

}

}

}

RadioButton rbApproved = (RadioButton)e.Row.FindControl("rbApproved");

RadioButton rbRejected = (RadioButton)e.Row.FindControl("rbRejected");

Label lblApprovalStatus = (Label)e.Row.FindControl("lblApprovalStatus");

rbApproved.Enabled = true;

rbRejected.Enabled = true;

if ((string)lblApprovalStatus.Text == "Approved")

{

rbApproved.Checked = true;

}

else if ((string)lblApprovalStatus.Text == "Rejected")

{

rbRejected.Checked = true;

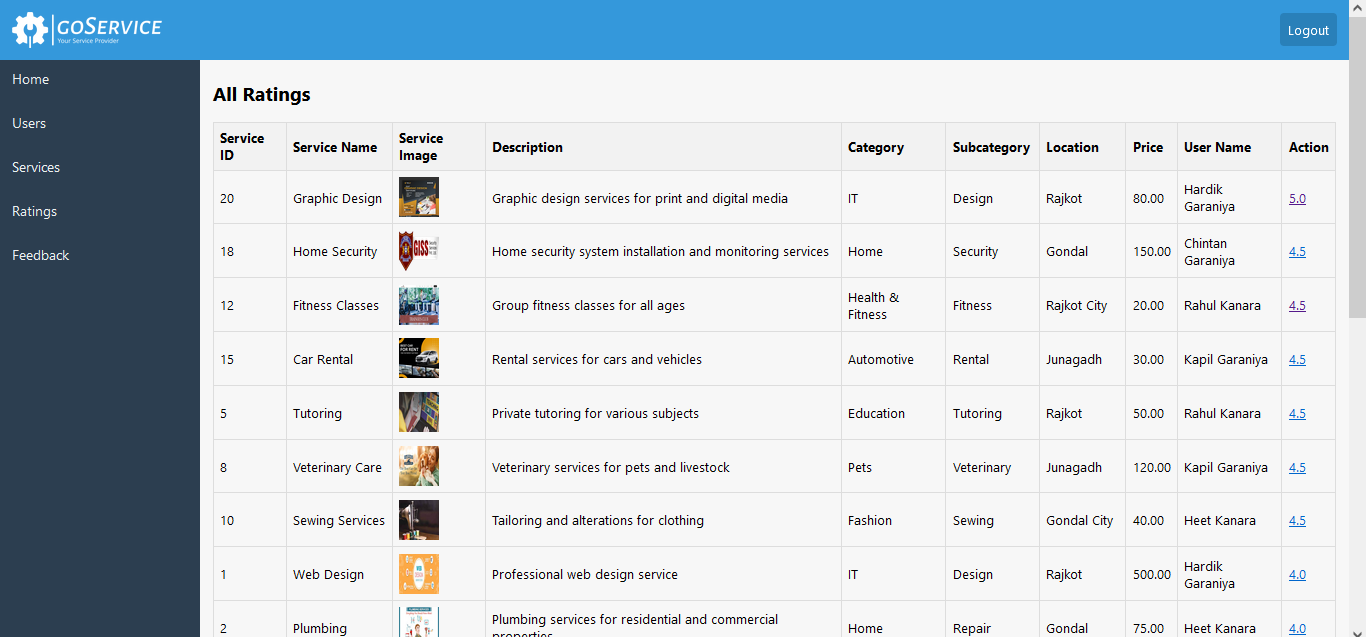
}

}

}

}

**All Ratings**



**:=> AdminRatings.aspx**

<%@ Page Language="C#" MasterPageFile="~/Admin/AdminMasterPage.master" AutoEventWireup="true" CodeFile="AdminRatings.aspx.cs" Inherits="Admin\_AdminRatings" Title="Admin Ratings" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<asp:GridView ID="GridViewServices" runat="server" AutoGenerateColumns="False" DataKeyNames="ServiceID">

<Columns>

<asp:BoundField DataField="ServiceID" HeaderText="Service ID" ReadOnly="true" SortExpression="ServiceID" />

<asp:BoundField DataField="ServiceName" HeaderText="Service Name" SortExpression="ServiceName" />

<asp:TemplateField HeaderText="Service Image">

<ItemTemplate>

<asp:Image ID="imgService" runat="server" Width="50px" Height="50px" ImageUrl='<%# ResolveUrl(Eval("ImagePath").ToString()) %>' AlternateText="Service Image" />

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="Description" HeaderText="Description" SortExpression="Description" />

<asp:BoundField DataField="Category" HeaderText="Category" SortExpression="Category" />

<asp:BoundField DataField="Subcategory" HeaderText="Subcategory" SortExpression="Subcategory" />

<asp:BoundField DataField="Location" HeaderText="Location" SortExpression="Location" />

<asp:BoundField DataField="Price" HeaderText="Price" SortExpression="Price" />

<asp:BoundField DataField="UserName" HeaderText="User Name" SortExpression="UserName" />

<asp:TemplateField HeaderText="Action">

<ItemTemplate>

<asp:HyperLink ID="lnkShowRatings" runat="server" NavigateUrl='<%# "~/Admin/AdminShowRating.aspx?ServiceID=" + Eval("ServiceID") %>' Text="Show Ratings" />

</ItemTemplate>

</asp:TemplateField>

</Columns>

</asp:GridView>

</asp:Content>

**:=> AdminRatings.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public partial class Admin\_AdminRatings : System.Web.UI.Page

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

LoadServices();

}

}

protected void LoadServices()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT s.ServiceID, s.ServiceName, s.Description, s.Category, s.Subcategory, s.Location, s.Price, u.UserName, s.ApprovalStatus, s.ImagePath, " +

"(SELECT AVG(CAST(RatingValue AS FLOAT)) FROM Rating WHERE ServiceID = s.ServiceID) AS OverallRating " +

"FROM Services s INNER JOIN [User] u ON s.ProviderID = u.UserID " +

"ORDER BY OverallRating DESC";

SqlCommand cmd = new SqlCommand(query, connection);

try

{

connection.Open();

SqlDataReader reader = cmd.ExecuteReader();

GridViewServices.DataSource = reader;

GridViewServices.DataBind();

}

catch (Exception ex)

{

Console.WriteLine("An error occurred: " + ex.Message);

}

}

}

protected void btnShowRatings\_Command(object sender, CommandEventArgs e)

{

if (e.CommandName == "ShowRatings")

{

int serviceID = Convert.ToInt32(e.CommandArgument);

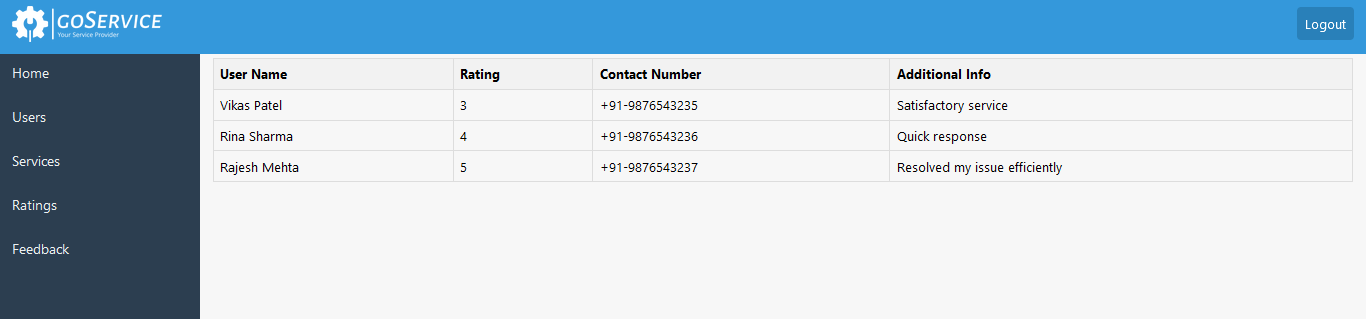
Response.Redirect("AdminShowRating.aspx?ServiceID={ServiceID}");

}

}

}

**Show Rating**



**:=>** **AdminShowRating.aspx**

<%@ Page Language="C#" MasterPageFile="~/Admin/AdminMasterPage.master" AutoEventWireup="true" CodeFile="AdminShowRating.aspx.cs" Inherits="Admin\_AdminShowRating" Title="Admin Show Ratings" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<asp:GridView ID="GridViewRatings" runat="server" AutoGenerateColumns="False">

<Columns>

<asp:BoundField DataField="UserName" HeaderText="User Name" SortExpression="UserName" />

<asp:BoundField DataField="RatingValue" HeaderText="Rating" SortExpression="RatingValue" />

<asp:BoundField DataField="ContactNumber" HeaderText="Contact Number" SortExpression="ContactNumber" />

<asp:BoundField DataField="AdditionalInfo" HeaderText="Additional Info" SortExpression="AdditionalInfo" />

</Columns>

</asp:GridView>

</asp:Content>

**:=>** **AdminShowRating.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public partial class Admin\_AdminShowRating : System.Web.UI.Page

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

LoadRatings();

}

}

protected void LoadRatings()

{

if (Request.QueryString["ServiceID"] != null)

{

int serviceID;

if (int.TryParse(Request.QueryString["ServiceID"], out serviceID))

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT \* FROM Rating WHERE ServiceID = @ServiceID";

SqlCommand cmd = new SqlCommand(query, connection);

cmd.Parameters.AddWithValue("@ServiceID", serviceID);

try

{

connection.Open();

GridViewRatings.DataSource = cmd.ExecuteReader();

GridViewRatings.DataBind();

}

catch (Exception ex)

{

Console.WriteLine("An error occurred: " + ex.Message);

}

}

}

else

{

Response.Redirect("~/ErrorPage.aspx");

}

}

else

{

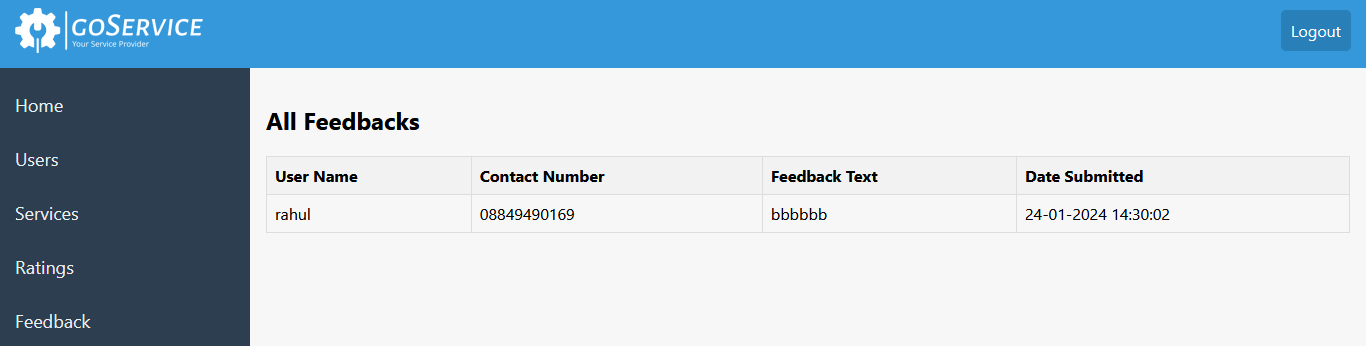
Response.Redirect("~/ErrorPage.aspx");

}

}

}

**All Feedbacks**



**:=> AdminFeedbacks.aspx**

<%@ Page Language="C#" MasterPageFile="~/Admin/AdminMasterPage.master" AutoEventWireup="true" CodeFile="AdminFeedbacks.aspx.cs" Inherits="Admin\_AdminFeedbacks" Title="Feedbacks" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

</style>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<div class="feedback-container">

<h2>All Feedbacks</h2>

<asp:Repeater ID="rptFeedbacks" runat="server">

<HeaderTemplate>

<table class="feedback-table">

<tr>

<th>User Name</th>

<th>Contact Number</th>

<th>Feedback Text</th>

<th>Date Submitted</th>

</tr>

</HeaderTemplate>

<ItemTemplate>

<tr>

<td><%# Eval("UserName") %></td>

<td><%# Eval("ContactNumber") %></td>

<td><%# Eval("FeedbackText") %></td>

<td><%# Eval("DateSubmitted") %></td>

</tr>

</ItemTemplate>

<FooterTemplate>

</table>

</FooterTemplate>

</asp:Repeater>

</div>

</asp:Content>

**:=> AdminFeedbacks.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public partial class Admin\_AdminFeedbacks : System.Web.UI.Page

{

private string connectionString = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

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

{

if (!IsPostBack)

{

BindFeedbacks();

}

}

private void BindFeedbacks()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT UserName, ContactNumber, FeedbackText, DateSubmitted FROM Feedback";

SqlCommand cmd = new SqlCommand(query, connection);

try

{

connection.Open();

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

rptFeedbacks.DataSource = dt;

rptFeedbacks.DataBind();

}

catch (Exception ex)

{

}

}

}

}

**Admin Logout**

**:=> AdminLogout.aspx**

<%@ Page Language="C#" MasterPageFile="~/Admin/AdminMasterPage.master" AutoEventWireup="true" CodeFile="AdminLogout.aspx.cs" Inherits="Admin\_AdminLogout" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

</asp:Content>

**:=> AdminLogout.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

public partial class Admin\_AdminLogout : System.Web.UI.Page

{

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

{

Session.Clear();

Session.Abandon();

Response.Redirect("../Login.aspx");

}

}

**Testing**

**Importance of software Testing**

He importance of software testing and its impact on software cannot be underestimated. Software testing is a fundamental component of software quality assurance and represents a review of specification, design and coding. The greater visibility of software system and the code associated with software failure are motivating factors for planning, through testing. It is not uncommon for a software organization to spent 40% of its efforts on testing.

**Software Testing Fundamentals**

During testing the software engineering produces a series of test cases that are used to “rip apart” the software they have produced. Testing is the one step in the software process that can be seen by the developer as destructive instead of constructive. Software engineers are typically constructive people and testing requires them to overcome preconceived concepts of correctness and deal with conflicts when errors are identified.

**Black Box Testing**

Black box testing relates to the tests that are performed at the software interface. Although they are designed identify errors, black box tests are used to demonstrate that software function are operational; that inputs are correctly accepted and the output is correctly produced. A black box test considers elements of the system with little interest in the internal logical arrangement of the software. White box testing of the testing involve closer examination of procedural detail. Logical paths through the software are considered by providing test cases that exercise particular sets of conditions and / or loops. The status of the system can be identified at diverse points to establish if the expected status matches the actual status.

Black box testing tries to find errors in the following categories:

1. Incorrect or missing functions,

2. Interface errors,

3. Errors in data structures or external database access,

4. Performance errors, and

5. Initialization and termination errors.

**White Box Testing**

White box testing is a test case design approach that employees the control architecture of the procedural design to produce test cases. Using white box test approaches, the software engineering can produce test cases

1. Guarantee that all independent paths in a module have been exercised at least once.
2. Exercise all logical decisions.
3. Execute all loops at their boundaries and in their operational bounds. Exercise internal data structures to maintain their validity.

**Future Scope**

* We can make a premium subscription for users that can the website gets more users.
* We can make the chat facility to users and service providers that can they directly communicate.
* We can make the sms functionality that can service providers give massage when they service gets approval.
* We can make the show intrest page for user that can we analys which type of services much needed.

**Limitation**

* This System is temporary not provide message services.
* This system is temporary use from local vendors.

**Bibliography**

* **Books**

1. System Analysis and Design.

By: James A Sen.

2. Software Engineering.

By: Roger Pressman.

3. Professional ASP.NET 2.0

* **Web site**

[www.google.com](http://www.google.com)

[www.wikipedia.com](http://www.wikipedia.com)

**Special Thanks**

🡪 Prof J. M. Ardeshna

🡪 Prof. V. M. Isotiya

At last but not least, we would like to extend our thanks to all supports, including family members and as well as our classmates and external friends who have helped us implicitly or explicitly. No amount of words written here will suffice for our sense of gratitude towards all of them.