**Registration**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Linq.Expressions;

using Online\_Bankine\_Transaction.Repo;

using System.Xml.Linq;

namespace Online\_Bankine\_Transaction

{

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

{

SqlConnection con;// Declaration of SqlConnection object

SqlCommand cmd;// Declaration of SqlCommand object

SqlDataReader reader; // Declaration of SqlDataReader object

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

{

if(!IsPostBack)

{

// If the page is not being loaded in response to a postback

lblAccountNumber.Text = displayAccountNumber();// Call to displayAccountNumber method to set account number label text

}

}

// Method to generate the next available account number

string displayAccountNumber()

{

// Establishing connection to the database using the connection string from Common class

//string con = ConnectionString.BankingConnString.ToString();

con = new SqlConnection(Common.GetConnectionString());

// SQL command to select the next available account number by incrementing the maximum existing account number

cmd = new SqlCommand(@"Select 'ABC20220000' + CAST( MAX( CAST( SUBSTRING(AccountNumber, 12, 50 ) AS INT)) +1 AS VARCHAR)

AS AccountNumber from Account", con);

con.Open();// Opening database connection

reader =cmd.ExecuteReader(); // Executing SQL command and storing result in SqlDataReader

string accountNumber = string.Empty;// Initializing account number variable

while (reader.Read()) // Loop through each record in the result

{

accountNumber = reader["AccountNumber"].ToString();// Assigning account number value

}

reader.Close();

con.Close();

return accountNumber;// Returning the generated account number

}

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

{

try

{

// Establishing connection to the database using the connection string from Common class

con = new SqlConnection(Common.GetConnectionString());

SqlCommand cmdSelect = new SqlCommand(@"SELECT Username FROM Account WHERE Username = @Username");

// Adding parameter to SQL command to prevent SQL injection

cmdSelect.Parameters.AddWithValue("@Username", txtUsername.Text);

con.Open();

cmdSelect.Connection = con;

cmdSelect.ExecuteNonQuery();

reader = cmdSelect.ExecuteReader();

if (reader.Read()) // If a record with the same username already exists

{

error.InnerText = "User name already exist. ";

}

else {

int securityID = ddlSecurityQuestion.SelectedIndex;

securityID += 1;

con = new SqlConnection(Common.GetConnectionString());

cmd = new SqlCommand(@"Insert into Account(AccountNumber, AccountType, Username,Gender,Email,Address,SecurityQuestionId,Answer,Amount,Password)

values(@AccountNumber, @AccountType, @Username,@Gender,@Email,@Address,@SecurityQuestionId,@Answer, @Amount, @Password)", con);

con.Open();

// Adding parameters to SQL command to prevent SQL injection

cmd.Parameters.AddWithValue("@AccountNumber", lblAccountNumber.Text);

cmd.Parameters.AddWithValue("@AccountType", lblAccountType.Text);

cmd.Parameters.AddWithValue("@Username", txtUsername.Text.Trim());

cmd.Parameters.AddWithValue("@Gender", ddlGender.Text);

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

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

cmd.Parameters.AddWithValue("@SecurityQuestionId", securityID);

cmd.Parameters.AddWithValue("@Answer", txtAnswer.Text.Trim());

cmd.Parameters.AddWithValue("@Amount", Convert.ToInt32(txtAmount.Text));

cmd.Parameters.AddWithValue("@Password", txtPassword.Text.Trim());

cmd.Connection = con;

cmd.ExecuteNonQuery(); // Executing SQL command to insert new account into database

Response.Redirect("Login.aspx"); ; // Redirecting user to login page after successful registration

}

}

catch (Exception msg)// Catching any exceptions that occur during registration process

{

error.InnerText = msg.Message;// Displaying error message

}

try

{

// Establishing connection to the database using the connection string from Common class

con.Open();

int r = cmd.ExecuteNonQuery();

if (r > 0)

{

Response.Redirect("Login.aspx", false); // Redirecting user to login page

}

else

{

error.InnerText = "Invalid input.";

}

}

catch (Exception ex)

{

if (ex.Message.Contains("Violation of UNIQUE KEY constraint"))

{

error.InnerText = "User name already exist. ";

}

else

{

Response.Write("<script>alert('Error- " + ex.Message + " ');</script>");

}

}

finally

{

con.Close();

}

}

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

{

Response.Redirect("Login.aspx"); // Redirecting user to login page

}

protected void txtUsername\_TextChanged(object sender, EventArgs e)

{

}

protected void txtPassword\_TextChanged(object sender, EventArgs e)

{

}

protected void txtConfirmPassword\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

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

{

}

}

}

Login.aspx

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

using System.Runtime.Remoting.Messaging;

namespace Online\_Bankine\_Transaction

{

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

{

SqlConnection con; // Declaration of SqlConnection object

SqlCommand cmd; // Declaration of SqlCommand object

SqlDataReader reader; // Declaration of SqlDataReader object

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

{

}

// Event handler for Register button click, redirects to registration page

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

{

Response.Redirect("Registration.aspx");

}

protected void lbForgotPassword(object sender, EventArgs e)

{

if (txtUsername.Text.Trim() == string.Empty)

{

// Display error message if username field is empty

error.InnerText = "Invalid input.";

txtUsername.Focus();

}

else

{

// Store the username in session variable and redirect to ForgotPassword page

Session["forgotpassword"] = txtUsername.Text.Trim();

Response.Redirect("ForgotPassword.aspx");

}

}

protected void btnLoginClick1(object sender, EventArgs e)

{

// Store the username in session variable and redirect to ForgotPassword page

// int i = 0;

SqlConnection con = new SqlConnection(Common.GetConnectionString());

// Creating SQL command to select user based on username and password

SqlCommand cmd = new SqlCommand(@"Select \* from Account where Username=@Username and Password=@Password");

cmd.Parameters.AddWithValue("@Username", txtUsername.Text);

cmd.Parameters.AddWithValue("@Password", txtPassword.Text);

try

{

//con.Open();

reader = cmd.ExecuteReader();// Executing SQL command and storing result in SqlDataReader

bool isTrue = false;

while (reader.Read())// Loop through each record in the result

{

isTrue = true; // Set flag to true as user exists

Session["userId"] = reader["AccountId"].ToString();// Store user's account ID in session

}

if (isTrue) // If user exists

{

Response.Redirect("PerformTransaction.aspx", false);

}

else

{

error.InnerText = "Wrong username or password.";

}

}

catch (Exception ex) // Catching any exceptions that occur during login process

{

Response.Write("<script>alert('Error - " + ex.Message + " ');</script>"); // Displaying error message

}

finally

{

con.Close();// Displaying error message

}

}

protected void txtPassword\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

}

ForgotPassword.aspx

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Online\_Bankine\_Transaction

{

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

{

SqlConnection con; // Declaration of SqlConnection object

SqlCommand cmd; // Declaration of SqlCommand object

SqlDataReader reader; // Declaration of SqlDataReader object

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

{

if(!IsPostBack)

{

// If the page is not being loaded in response to a postback

getUserSecurityQuestion();

}

}

// Method to retrieve and display user's security question

void getUserSecurityQuestion()

{

if (Session["forgotpassword"] != null) // Check if forgotpassword session variable is set

{

con = new SqlConnection(Common.GetConnectionString());

cmd = new SqlCommand(@"Select Username,SecurityQuestionName, Answer from Account inner join SecurityQuestion on Account.SecurityQuestionId = SecurityQuestion.SecurityQuestionId where Username =@Username");

cmd.Parameters.AddWithValue("@Username", Session["forgotpassword"]);

con.Open();

cmd.Connection=con; cmd.ExecuteNonQuery();

try

{

//con.Open();

reader = cmd.ExecuteReader();// Execute SQL command and store result in SqlDataReader

bool isTrue = false;

while (reader.Read()) // Loop through each record in the result

{

isTrue = true; // Set flag to true as user exists

//lblUsername.Text = reader["Username"].ToString();

lblUsername.Text = Session["ForgotPassword"].ToString();// Display username

lblSecurityQuestion.Text = reader["SecurityQuestionName"].ToString();// Display security question

hdnAnswer.Value = reader["Answer"].ToString() ;

}

if (!isTrue)// If user does not exist

{

error.InnerText = "Invalid input.";

}

}

catch (Exception ex)

{

Response.Write("<script>alert('Error - " + ex.Message + " ' );</script>");

}

finally // Executed regardless of whether there's an exception or not

{

if(reader!= null)

{

reader.Close(); // Close SqlDataReader

}

}

con.Close();// Close SqlDataReader

}

}

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

{

if(txtAnswer.Text.Trim() == hdnAnswer.Value) // Check if entered answer matches stored answer

{

Response.Redirect("ChangePassword.aspx"); // Redirect to ChangePassword page

}

else

{

error.InnerText = "Invalid inpput";// Display error message if answer is incorrect

}

}

protected void HiddenField1\_ValueChanged(object sender, EventArgs e)

{

}

// Event handler for "Cancel" button click, redirects to login page

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

{

Response.Redirect("Login.aspx"); // Redirect to login page

}

protected void hdnAnswer\_ValueChanged(object sender, EventArgs e)

{

}

protected void TextBox1\_TextChanged(object sender, EventArgs e)

{

}

protected void txtAnswer\_TextChanged(object sender, EventArgs e)

{

}

protected void LinkButton1\_Click(object sender, EventArgs e)

{

}

}

}

MyCredit.aspx

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Security.Cryptography;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Online\_Bankine\_Transaction

{

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

{

SqlConnection con;// Declaration of SqlConnection object

SqlCommand cmd; // Declaration of SqlCommand object

SqlDataAdapter sda; // Declaration of SqlDataAdapter object

DataTable dt; // Declaration of DataTable object

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

{

// Check if the page is being loaded in response to a postback

if (IsPostBack)

{

// Check if the userId session variable is not set, indicating the user is not logged in

if (Session["userId"] == null)

{

// Redirect the user to the login page

Response.Redirect("Login.aspx");

}

// Call the getMyCredits method to fetch and display the user's credits

getMyCredits();

}

}

void getMyCredits()

{

try

{

con = new SqlConnection(Common.GetConnectionString());// Establishing connection to the databas

cmd = new SqlCommand(@"Select a.AccountNumber ,a.Username,t.Amount,t.Remarks from [Transaction] t inner join Account a on t.ReceiverAccountId =a.AccountId where t.ReceiverAccountId = @ReceiverAccountId", con);// SQL query to retrieve user's credits

cmd.Parameters.AddWithValue("@ReceiverAccountId", Session["userId"]); // Adding parameter to the SQL query

sda = new SqlDataAdapter(cmd); // Initializing SqlDataAdapter with SqlCommand

dt = new DataTable(); // Initializing DataTable

sda.Fill(dt); // Filling DataTable with data from SqlDataAdapter

gvMyCredits.DataSource = dt; // Setting GridView's data source to the DataTable

gvMyCredits.DataBind(); // Binding the data to the GridView

}

catch (Exception ex) // Catching any exceptions that occur during database operations

{

// Displaying error message in a script block

Response.Write("<script>alert(' " + ex.Message + " ');</script>");

}

}

}

}

myDebits

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Security.Cryptography;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Online\_Bankine\_Transaction

{

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

{

SqlConnection con; // Declaration of SqlConnection object

SqlCommand cmd; // Declaration of SqlCommand object

SqlDataAdapter sda; // Declaration of SqlDataAdapter object

DataTable dt; // Declaration of DataTable object

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

{

// Check if the page is being loaded in response to a postback

if (IsPostBack)

{

// Check if the userId session variable is not set, indicating the user is not logged in

if (Session["userId"] == null)

{

// Redirect the user to the login page if not logged in

Response.Redirect("Login.aspx");

}

// Call the getMyDebits method to fetch and display the user's debits

getMyDebits();

}

}

// Method to fetch and display the user's debits

void getMyDebits()

{

try

{

con = new SqlConnection(Common.GetConnectionString()); // Establish connection to the database

cmd = new SqlCommand(@"Select a.AccountNumber ,a.Username,t.Amount,t.Remarks from [Transaction] t inner join Account a on t.ReceiverAccountId =a.AccountId where t.SenderAccountId = @SenderAccountId", con); // Establish connection to the database

cmd.Parameters.AddWithValue("@SenderAccountId", Session["userId"]); // Adding parameter to the SQL query

sda = new SqlDataAdapter(cmd); // Initializing SqlDataAdapter with SqlCommand

dt = new DataTable();// Initializing DataTable

sda.Fill(dt); // Filling DataTable with data from SqlDataAdapter

gvMyDebits.DataSource = dt; // Setting GridView's data source to the DataTable

gvMyDebits.DataBind(); // Binding the data to the GridView

}

catch (Exception ex) // Catching any exceptions that occur during database operations

{

// Displaying error message in a script block

Response.Write("<script>alert(' " + ex.Message + " ');</script>");

}

}

}

}

PerformanceTransaction.aspx

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Online\_Bankine\_Transaction

{

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

{

SqlConnection con; // Declaration of SqlConnection object

SqlCommand cmd; // Declaration of SqlCommand object

SqlDataAdapter sda; // Declaration of SqlDataAdapter object

DataTable dt; // Declaration of DataTable object

SqlDataReader dr; // Declaration of SqlDataReader object

SqlTransaction transaction = null; // Declaration of SqlTransaction object

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

{

if (!IsPostBack)

{

if (Session["userId"] == null)

{

Response.Redirect("login.aspx");

}

getAccountNumber();

}

}

void getAccountNumber()

{

try

{

con = new SqlConnection(Common.GetConnectionString()); // Establish connection to the database

cmd = new SqlCommand(@"Select AccountId ,AccountNumber from Account where AccountId = != @AccountId", con); // SQL query to retrieve account numbers excluding the user's own account

cmd.Parameters.AddWithValue("@AccountId", Session["userId"]); // Adding parameter to the SQL query

sda = new SqlDataAdapter(cmd); // Initializing SqlDataAdapter SqlCommand

dt = new DataTable(); // Initializing DataTable

sda.Fill(dt); // Filling DataTable with data from SqlDataAdapter

ddlPayeeAccountNumber.DataSource = dt; // Setting dropdown list's data source to the DataTable

ddlPayeeAccountNumber.DataTextField = "AccountNumber"; // Setting the data field for text display

ddlPayeeAccountNumber.DataValueField = "AccoumtId"; // Setting the data field for value retrieval

ddlPayeeAccountNumber.DataBind(); // Binding the data to the dropdown list

}

catch (Exception ex) // Catching any exceptions that occur during database operations

{

// Displaying error message in a script block

Response.Write("<script>alert(' " + ex.Message + " ');</script>");

}

}

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

{

if (Session["userId"] != null) // Check if the user is logged in

{

con = new SqlConnection(Common.GetConnectionString()); // Establish connection to the database

con.Open(); // Open the database connection

try

{

int r = 0; // Initialize variable to track the number of affected rows

Utils utils = new Utils(); // Creating an instance of the Utils class

int balanceAmount = utils.accountBalance(Convert.ToInt32(Session["userId"])); // Retrieve user's account balance

if (Convert.ToInt32(txtAmount.Text.Trim()) <= balanceAmount) // Check if the transaction amount is less than or equal to the account balance

{

transaction = con.BeginTransaction();

cmd = new SqlCommand(@"Insert into [Transaction](SenderAccountId, ReceiverAccountId, MobileNo, Amount, TransactionType, Remarks)

values (@SenderAccountId, @ReceiverAccountId, @MobileNo, @Amount, @TransactionType, @Remarks)", con, transaction); // SQL command to insert transaction details

cmd.Parameters.AddWithValue("@SenderAccountId", Session["userId"]); // Adding parameter for sender account ID

cmd.Parameters.AddWithValue("@ReceiverAccountId", ddlPayeeAccountNumber.SelectedValue); // Adding parameter for receiver account ID

cmd.Parameters.AddWithValue("@MobileNo", txtMobileNumber.Text.Trim()); // Adding parameter for mobile number

cmd.Parameters.AddWithValue("@Amount", txtAmount.Text.Trim()); // Adding parameter for transaction amount

cmd.Parameters.AddWithValue("@TransactionType", "DR"); // Adding parameter for transaction type (debit)

cmd.Parameters.AddWithValue("@Remarks", txtRemarks.Text.Trim()); // Adding parameter for transaction remarks

r = cmd.ExecuteNonQuery(); // Executing SQL command and getting the number of affected rows

// Update sender's account balance

UpdateSenderAccountBalance(Convert.ToInt32(Session["userId"]), balanceAmount, Convert.ToInt32(txtAmount.Text.Trim()), con, transaction);

UpdateReceiverAccountBalance(Convert.ToInt32(ddlPayeeAccountNumber.SelectedValue), Convert.ToInt32(txtAmount.Text.Trim()), con, transaction);

// Update receiver's account balance

transaction.Commit();// Commit the database transaction

r = 1;

if (r > 0)

{

// Redirect the user to the mydebits page

Response.Redirect("mydebits.aspx", false);

}

else

{

error.InnerText = "invalid Input";

}

}

else

{

error.InnerText = "Invalid Input";

}

}

catch (Exception)

{

try

{

transaction.Rollback();

}

catch (Exception ex)

{

Response.Write("<script>alert('" + ex.Message + "');</script>");

}

}

finally

{

con.Close();

}

}

}

void UpdateSenderAccountBalance(int \_senderId, int \_dbAmount, int \_amount, SqlConnection SqlConnection, SqlTransaction sqlTransaction)

{

try

{

// Check if there are sufficient funds in the sender's account

if (\_dbAmount >= \_amount) // Deduct the transaction amount from the current balance

{

\_dbAmount -= \_amount;

// Update the sender's account balance in the database

cmd = new SqlCommand("Update Account set Amount =@Amount where where AccountId= @AccountId", SqlConnection, sqlTransaction);

cmd.Parameters.AddWithValue("\_dbAmount", \_dbAmount);

cmd.Parameters.AddWithValue("@AccountId", \_senderId);

cmd.ExecuteNonQuery();

}

}

catch (Exception ex)

{

// Display error message if an exception occurs during the database operation

Response.Write("<script.alert('" + ex.Message + "');</script>");

}

}

void UpdateReceiverAccountBalance(int \_receiverId, int \_amount, SqlConnection sqlConnection, SqlTransaction sqlTransaction)

{

int \_dbAmount = 0; // Variable to store the receiver's account balance

// Retrieve the receiver's current account balance from the database

cmd = new SqlCommand("Select Amount from Account where AccountId =@AccountId", sqlConnection, sqlTransaction);

cmd.Parameters.AddWithValue("@AccountId", \_receiverId);

try

{

dr = cmd.ExecuteReader(); // Execute the SQL command to retrieve the account balance

while (dr.Read())

{

\_dbAmount = (int)dr["Amount"]; // Store the retrieved account balance

// Add the transaction amount to the receiver's account balance

\_dbAmount += \_amount;

// Update the receiver's account balance in the database

cmd = new SqlCommand("Update Account set Amount = @Amount where AccountId = @AccountId", sqlConnection, sqlTransaction);

cmd.Parameters.AddWithValue("@Amount", \_dbAmount);

cmd.Parameters.AddWithValue("@AccountId", \_receiverId);

cmd.ExecuteNonQuery(); // Execute the SQL command to update the receiver's account balance

}

dr.Close(); // Close the SqlDataReader

}

catch(Exception ex)

{

Response.Write("<script>alert('" + ex.Message + "');<script>");

}

}

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

{

Response.Redirect("PerformTransaction.aspx");

}

}

}