Backoffice Approvals

Credit Card Approval

Load button click:

protected void LoadClientDocuments(object sender, EventArgs e)

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

int login = 0;

if (txtAccno.Value != "")

{

login = Convert.ToInt32(txtAccno.Value);

}

if (txtAccno.Value != "")

{

txtCardNo.Value = string.Empty;

txtEditCardNo.Value = string.Empty;

txtEditName.Value = string.Empty;

txtFullName.Value = string.Empty;

txtname.Value = string.Empty;

div\_DocList.InnerHtml = string.Empty;

DataTable dt = cr.GetCreditCardDetails(login); // SP\_GetCreditCardDetails

StringBuilder html = new StringBuilder();

if (dt.Rows.Count > 0)

{

txtFullName.Value = dt.Rows[0]["Name"].ToString();

txtname.Value = dt.Rows[0]["Name"].ToString();

if (dt.Rows[0]["Id"].ToString() != "0")

{

html.Append("<table class='table table-striped projects'>");

html.Append("<thead><tr>");

foreach (DataColumn column in dt.Columns)

{

if (column.ColumnName != "Id")

{

html.Append("<th>");

html.Append(column.ColumnName);

html.Append("</th>");

}

}

html.Append("</tr></thead><tbody>");

foreach (DataRow row in dt.Rows)

{

html.Append("<tr>");

foreach (DataColumn column in dt.Columns)

{

if (column.ColumnName != "Id")

{

if (column.ColumnName != "Status")

{

if (column.ColumnName == "imageName")

{

html.Append("<td width='150px'><a>");

html.Append(row[column.ColumnName]);

html.Append("</a></td>");

}

else

{

html.Append("<td><a>");

html.Append(row[column.ColumnName]);

html.Append("</a></td>");

}

}

else

{

if (row[column.ColumnName].ToString() == "Pending")

{

html.Append("<td><button type='button' class='btn btn-warning btn-xs'>" + row[column.ColumnName] + "</button></td>");

}

else if (row[column.ColumnName].ToString() == "Rejected")

{

html.Append("<td><button type='button' class='btn btn-danger btn-xs'>" + row[column.ColumnName] + "</button></td>");

}

else

{

html.Append("<td><button type='button' class='btn btn-success btn-xs'>" + row[column.ColumnName] + "</button></td>");

}

}

}

}

string appstr = "document.getElementById(" + "&#039;" + "hdndownloadID" + "&#039;).value=&#039;" + row["Id"].ToString() + "&#039;;document.getElementById(" + "&#039;" + "hdnFourDigit" + "&#039;).value=&#039;" + row["CardNo"].ToString() + "&#039;;document.getElementById(" + "&#039;" + "hdnCardType" + "&#039;).value=&#039;" + row["Type"].ToString() + "&#039;";

string host = HttpContext.Current.Request.Url.Host;

string getimage = "";

string getbackimage = "";

string editclick = "";

getimage = "window.open(&#039;http://185.4.178.133:13956/getCreditCardDetails.aspx?login=" + login.ToString() + "&orderno=" + row["Id"].ToString() + "&side=0&#039;);";

getbackimage = "window.open(&#039;http://185.4.178.133:13956/getCreditCardDetails.aspx?login=" + login.ToString() + "&orderno=" + row["Id"].ToString() + "&side=1&#039;);";

if (row["Status"].ToString() == "Approved")

editclick = "document.getElementById(&#039;txtEditCardNo&#039;).value=&#039;" + row["CardNo"].ToString() + "&#039;; document.getElementById(&#039;txtEditCardNo&#039;).setAttribute(&#039;readonly&#039;,&#039;readonly&#039;); document.getElementById(&#039;txtEditName&#039;).value=&#039;" + row["Name"].ToString() + "&#039;; document.getElementById(&#039;ddlyear&#039;).value=&#039;" + row["ExpYear"].ToString() + "&#039;; document.getElementById(&#039;ddledMonth&#039;).value=&#039;" + row["ExpMonth"].ToString() + "&#039;; document.getElementById(" + "&#039;" + "hdndownloadID" + "&#039;).value=&#039;" + row["Id"].ToString() + "&#039;; document.getElementById(&#039;ddlED\_Type&#039;).value=&#039;" + row["Type"].ToString() + "&#039;";

else

editclick = "document.getElementById(&#039;txtEditCardNo&#039;).value=&#039;" + row["CardNo"].ToString() + "&#039;; document.getElementById(&#039;txtEditCardNo&#039;).removeAttribute(&#039;readonly&#039;); document.getElementById(&#039;txtEditName&#039;).value=&#039;" + row["Name"].ToString() + "&#039;; document.getElementById(&#039;ddlyear&#039;).value=&#039;" + row["ExpYear"].ToString() + "&#039;; document.getElementById(&#039;ddledMonth&#039;).value=&#039;" + row["ExpMonth"].ToString() + "&#039;; document.getElementById(" + "&#039;" + "hdndownloadID" + "&#039;).value=&#039;" + row["Id"].ToString() + "&#039;; document.getElementById(&#039;ddlED\_Type&#039;).value=&#039;" + row["Type"].ToString() + "&#039;";

int orderno = Convert.ToInt32(row["Id"].ToString());

string orderno1 = (Request.Form["hdndownloadID"]);

login = Convert.ToInt32(login.ToString());

html.Append("<td><a onclick='" + getimage + "' class='btn btn-primary btn-xs'><i class='fa fa-arrow-down'></i> View Card Front </a><a onclick='" + getbackimage + "' class='btn btn-primary btn-xs'><i class='fa fa-arrow-down'></i> View Card Back </a><a data-toggle='modal' data-target='#dv\_approve\_Confirm' onclick='" + appstr + "' class='btn btn-success btn-xs'><i class='fa fa-check-square-o'></i> Approve </a><a data-toggle='modal' data-target='#dv\_revert\_comfirm' onclick='" + appstr + "' class='btn btn-danger btn-xs'><i class='fa fa-undo'></i> Reject </a><a class='btn btn-dark btn-xs' data-toggle='modal' data-target='#dv\_addDetails' onclick='" + editclick + "'><i class='fa fa-pencil'></i> Edit </a></td></tr>");

}

html.Append("</tbody></table>");

div\_DocList.InnerHtml = html.ToString();

}

}

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please Enter Account Number.')", true);

txtAccno.Focus();

return;

}

}

catch (Exception ex)

{

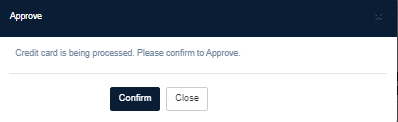
ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

Approve button click:



Confirm:

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

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

CreditCard\_DataAccess crdata = new CreditCard\_DataAccess();

CreditCard\_Info crinfo = new CreditCard\_Info();

int Result = 0;

int login;

if (txtAccno.Value == "")

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter Login to continue.')", true);

return;

}

SalesLead\_DataAccess salesda = new SalesLead\_DataAccess();

SalesLeadsInfo info1 = new SalesLeadsInfo();

info1 = salesda.GetAcctBasicDts(txtAccno.Value); // SP\_GetAcctBasicDts

login = Convert.ToInt32(txtAccno.Value);

string orderno = (Request.Form["hdndownloadID"]);

string fourdigit = Request.Form["hdnFourDigit"];

string cardtype= Request.Form["hdnCardType"];

if (fourdigit != "" && fourdigit != "0")

{

Result = crdata.UpdateCreditCardStatus(login, Convert.ToInt32(orderno), "Approved", Convert.ToInt32(Session["UID"]));// SP\_VerifyCreditCard

if (Result > 0)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Credit Card Updated and Approved.')", true);

EmailFile emf = new EmailFile();

string em = emf.CreditCardApproved(info1, fourdigit, cardtype);

LoadClientDocuments(this, null);

return;

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please check the fields')", true);

return;

}

}

}

catch (Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

public string CreditCardApproved(SalesLeadsInfo info, string fourdigit, string cardtype)

{

string res = string.Empty;

try

{

string msg = string.Empty;

string arrtitle = string.Empty;

using (WebClient webClient = new System.Net.WebClient())

{

WebClient objWebClient = new WebClient();

var strcontent = "";

if (cardtype == "Credit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CreditCardApproved.html");

else if (cardtype == "Debit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/DebitCardApproved.html");

else if (cardtype == "Crypto")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CryptoCardApproved.html");

msg = Convert.ToString(strcontent);

}

msg = msg.Replace("[FourDigits]", fourdigit);

if (info.Title != "")

msg = msg.Replace("[NameWithTitle]", info.Title + " " + info.Name);

else

msg = msg.Replace("[NameWithTitle]", info.Name);

string MailServer = "";

string ToEmail = info.Email1;

string ToName = info.Name;

string FromEmail = "";

string FromName = "";

string Subject = "";

if (cardtype == "Credit")

Subject = "Credit Card Approved";

else if (cardtype == "Debit")

Subject = "Debit Card Approved";

else if (cardtype == "Crypto")

Subject = "Crypto Card Approved";

SmtpClient o\_Client;

MailServer = M\_MailServer;

FromEmail = M\_SalesFromEmail;

FromName = M\_FromName;

o\_Client = new SmtpClient(MailServer);

o\_Client.Credentials = new System.Net.NetworkCredential(M\_SalesFromEmail, M\_SalesEmailPassword);

o\_Client.Port = M\_Port;

o\_Client.Timeout = 25000;

o\_Client.EnableSsl = true;

o\_Client.Host = M\_MailServer;

MailMessage o\_Message = new MailMessage();

o\_Message.From = new MailAddress(FromEmail, FromName);

int iname = 0;

foreach (string to in ToEmail.Split(';'))

{

if (ToName.Contains(';'))

o\_Message.To.Add(new MailAddress(to, ToName.Split(';')[iname]));

else

o\_Message.To.Add(new MailAddress(to, ToName));

iname++;

}

o\_Message.Bcc.Add("crm@trusttc.com");

o\_Message.Bcc.Add("magt@trusttc.com");

o\_Message.Bcc.Add("backoffice@trusttc.com");

o\_Message.Subject = Subject;

o\_Message.IsBodyHtml = true;

o\_Message.Body = msg;

System.Net.ServicePointManager.ServerCertificateValidationCallback = delegate (object s,

System.Security.Cryptography.X509Certificates.X509Certificate certificate,

System.Security.Cryptography.X509Certificates.X509Chain chain,

System.Net.Security.SslPolicyErrors sslPolicyErrors)

{

return true;

};

o\_Client.Send(o\_Message);

o\_Message.Dispose();

res = "Email Sent successfully";

}

catch (Exception ex)

{

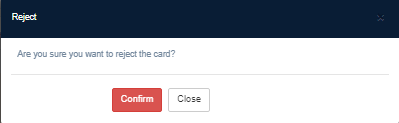
res = ex.Message;

}

return res;

}

Reject:



Confirm click:

public void Revert\_Click(object sender, EventArgs e)

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

SalesLead\_DataAccess salesda = new SalesLead\_DataAccess();

SalesLeadsInfo info1 = new SalesLeadsInfo();

info1 = salesda.GetAcctBasicDts(txtAccno.Value);

int orderno = 0;

int login = 0;

int result = 0;

if (txtAccno.Value != "")

{

login = Convert.ToInt32(txtAccno.Value);

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter Login to continue.')", true);

return;

}

string fourdigit = Request.Form["hdnFourDigit"];

orderno = Convert.ToInt32(Request.Form["hdndownloadID"]);

string cardtype = Request.Form["hdnCardType"];

result = cr.UpdateCreditCardStatus(login, orderno, "Rejected", Convert.ToInt32(Session["UID"]));

// SP\_VerifyCreditCard

EmailFile emf = new EmailFile();

string em = emf.CreditCardRejected(info1, fourdigit, cardtype);

LoadClientDocuments(this, null);

}

catch (Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

public string CreditCardRejected(SalesLeadsInfo info, string fourdigits, string cardtype)

{

string res = string.Empty;

try

{

string msg = string.Empty;

string arrtitle = string.Empty;

using (WebClient webClient = new System.Net.WebClient())

{

WebClient objWebClient = new WebClient();

var strcontent = "";

if (cardtype == "Credit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CreditCardRejected.html");

else if (cardtype == "Debit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/DebitCardRejected.html");

else if (cardtype == "Crypto")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CryptoCardRejected.html");

msg = Convert.ToString(strcontent);

}

msg = msg.Replace("[FourDigits]", fourdigits.ToString());

if (info.Title != "")

msg = msg.Replace("[NameWithTitle]", info.Title + " " + info.Name);

else

msg = msg.Replace("[NameWithTitle]", info.Name);

string MailServer = "";

string ToEmail = info.Email1;

string ToName = info.Name;

string FromEmail = "";

string FromName = "";

string Subject = "";

if (cardtype == "Credit")

Subject = "Credit Card Rejected";

else if (cardtype == "Debit")

Subject = "Debit Card Rejected";

else if (cardtype == "Crypto")

Subject = "Crypto Card Rejected";

SmtpClient o\_Client;

MailServer = M\_MailServer;

FromEmail = M\_TransactionsFromEmail;

FromName = M\_FromName;

o\_Client = new SmtpClient(MailServer);

o\_Client.Credentials = new System.Net.NetworkCredential(M\_TransactionsFromEmail, M\_TransactionsEmailPassword);

o\_Client.Port = M\_Port;

o\_Client.Timeout = 25000;

o\_Client.EnableSsl = true;

o\_Client.DeliveryMethod = SmtpDeliveryMethod.Network;

o\_Client.Host = M\_MailServer;

MailMessage o\_Message = new MailMessage();

o\_Message.From = new MailAddress(FromEmail, FromName);

int iname = 0;

foreach (string to in ToEmail.Split(';'))

{

if (ToName.Contains(';'))

o\_Message.To.Add(new MailAddress(to, ToName.Split(';')[iname]));

else

o\_Message.To.Add(new MailAddress(to, ToName));

iname++;

}

o\_Message.Bcc.Add("crm@trusttc.com");

o\_Message.Bcc.Add("magt@trusttc.com");

o\_Message.Bcc.Add("backoffice@trusttc.com");

o\_Message.Subject = Subject;

o\_Message.IsBodyHtml = true;

o\_Message.Body = msg;

System.Net.ServicePointManager.ServerCertificateValidationCallback = delegate (object s,

System.Security.Cryptography.X509Certificates.X509Certificate certificate,

System.Security.Cryptography.X509Certificates.X509Chain chain,

System.Net.Security.SslPolicyErrors sslPolicyErrors)

{

return true;

};

o\_Client.Send(o\_Message);

o\_Message.Dispose();

res = "Email Sent successfully";

}

catch (Exception ex)

{

res = ex.Message;

}

return res;

}

public string CreditCardRejected(SalesLeadsInfo info, string fourdigits, string cardtype)

{

string res = string.Empty;

try

{

string msg = string.Empty;

string arrtitle = string.Empty;

using (WebClient webClient = new System.Net.WebClient())

{

WebClient objWebClient = new WebClient();

var strcontent = "";

if (cardtype == "Credit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CreditCardRejected.html");

else if (cardtype == "Debit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/DebitCardRejected.html");

else if (cardtype == "Crypto")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CryptoCardRejected.html");

msg = Convert.ToString(strcontent);

}

msg = msg.Replace("[FourDigits]", fourdigits.ToString());

if (info.Title != "")

msg = msg.Replace("[NameWithTitle]", info.Title + " " + info.Name);

else

msg = msg.Replace("[NameWithTitle]", info.Name);

string MailServer = "";

string ToEmail = info.Email1;

string ToName = info.Name;

string FromEmail = "";

string FromName = "";

string Subject = "";

if (cardtype == "Credit")

Subject = "Credit Card Rejected";

else if (cardtype == "Debit")

Subject = "Debit Card Rejected";

else if (cardtype == "Crypto")

Subject = "Crypto Card Rejected";

SmtpClient o\_Client;

MailServer = M\_MailServer;

FromEmail = M\_TransactionsFromEmail;

FromName = M\_FromName;

o\_Client = new SmtpClient(MailServer);

o\_Client.Credentials = new System.Net.NetworkCredential(M\_TransactionsFromEmail, M\_TransactionsEmailPassword);

o\_Client.Port = M\_Port;

o\_Client.Timeout = 25000;

o\_Client.EnableSsl = true;

o\_Client.DeliveryMethod = SmtpDeliveryMethod.Network;

o\_Client.Host = M\_MailServer;

MailMessage o\_Message = new MailMessage();

o\_Message.From = new MailAddress(FromEmail, FromName);

int iname = 0;

foreach (string to in ToEmail.Split(';'))

{

if (ToName.Contains(';'))

o\_Message.To.Add(new MailAddress(to, ToName.Split(';')[iname]));

else

o\_Message.To.Add(new MailAddress(to, ToName));

iname++;

}

o\_Message.Bcc.Add("crm@trusttc.com");

o\_Message.Bcc.Add("magt@trusttc.com");

o\_Message.Bcc.Add("backoffice@trusttc.com");

o\_Message.Subject = Subject;

o\_Message.IsBodyHtml = true;

o\_Message.Body = msg;

System.Net.ServicePointManager.ServerCertificateValidationCallback = delegate (object s,

System.Security.Cryptography.X509Certificates.X509Certificate certificate,

System.Security.Cryptography.X509Certificates.X509Chain chain,

System.Net.Security.SslPolicyErrors sslPolicyErrors)

{

return true;

};

o\_Client.Send(o\_Message);

o\_Message.Dispose();

res = "Email Sent successfully";

}

catch (Exception ex)

{

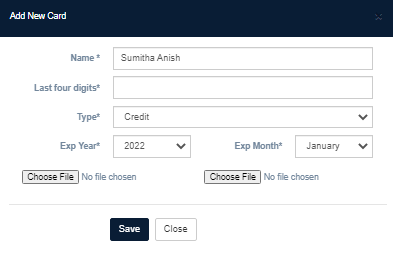
res = ex.Message;

}

return res;

}

Add Card:



Save Click:

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

{

CreditCard\_DataAccess crdata = new CreditCard\_DataAccess();

CreditCard\_Info crinfo = new CreditCard\_Info();

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

if (crdata.getCreditCardCount(Convert.ToInt32(txtAccno.Value)) >= 5)

// SP\_GetCreditCardCount

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Cannot upload more than five Credit Cards.')", true);

return;

}

if (txtname.Value == "")

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter the Name as in Credit Card')", true);

return;

}

if (txtCardNo.Value == "" || txtCardNo.Value == "0" || txtCardNo.Value.Length < 4 || Convert.ToInt32(txtCardNo.Value) == 0)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter the last four digits of the credit card no.')", true);

return;

}

if (ddlMonth.SelectedIndex.ToString() == "")

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter the expiration month.')", true);

return;

}

if (cmbYear.SelectedValue == "")

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter the expiration year.')", true);

return;

}

if (FileUpload1.FileName != null && FileUpload1.FileName != "" && FileUploadbk.FileName != null && FileUploadbk.FileName != "")

{

string filePath = FileUpload1.PostedFile.FileName;

string filename = Path.GetFileName(filePath);

string ext = Path.GetExtension(filename);

string contenttype = String.Empty;

string filePathbk = FileUploadbk.PostedFile.FileName;

string filenamebk = Path.GetFileName(filePathbk);

string extbk = Path.GetExtension(filenamebk);

string contenttypebk = String.Empty;

switch (ext)

{

case ".doc":

contenttype = "application/vnd.ms-word";

break;

case ".docx":

contenttype = "application/vnd.ms-word";

break;

case ".xls":

contenttype = "application/vnd.ms-excel";

break;

case ".xlsx":

contenttype = "application/vnd.ms-excel";

break;

case ".jpg":

contenttype = "image/jpg";

break;

case ".JPG":

contenttype = "image/jpg";

break;

case ".JPEG":

contenttype = "image/jpg";

break;

case ".jpeg":

contenttype = "image/jpg";

break;

case ".png":

contenttype = "image/png";

break;

case ".PNG":

contenttype = "image/png";

break;

case ".gif":

contenttype = "image/gif";

break;

case ".GIF":

contenttype = "image/gif";

break;

case ".bmp":

contenttype = "image/bmp";

break;

case ".BMP":

contenttype = "image/bmp";

break;

case ".pdf":

contenttype = "application/pdf";

break;

case ".PDF":

contenttype = "application/pdf";

break;

}

switch (extbk)

{

case ".doc":

contenttypebk = "application/vnd.ms-word";

break;

case ".docx":

contenttypebk = "application/vnd.ms-word";

break;

case ".xls":

contenttypebk = "application/vnd.ms-excel";

break;

case ".xlsx":

contenttypebk = "application/vnd.ms-excel";

break;

case ".jpg":

contenttypebk = "image/jpg";

break;

case ".JPG":

contenttypebk = "image/jpg";

break;

case ".JPEG":

contenttypebk = "image/jpg";

break;

case ".jpeg":

contenttypebk = "image/jpg";

break;

case ".png":

contenttypebk = "image/png";

break;

case ".PNG":

contenttypebk = "image/png";

break;

case ".gif":

contenttypebk = "image/gif";

break;

case ".GIF":

contenttypebk = "image/gif";

break;

case ".bmp":

contenttypebk = "image/bmp";

break;

case ".BMP":

contenttypebk = "image/bmp";

break;

case ".pdf":

contenttypebk = "application/pdf";

break;

case ".PDF":

contenttypebk = "application/pdf";

break;

}

if (contenttype != String.Empty && contenttypebk!=String.Empty)

{

Stream fs = FileUpload1.PostedFile.InputStream;

BinaryReader br = new BinaryReader(fs);

Byte[] bytes = br.ReadBytes((Int32)fs.Length);

Stream fsbk = FileUploadbk.PostedFile.InputStream;

BinaryReader brbk = new BinaryReader(fsbk);

Byte[] bytesbk = brbk.ReadBytes((Int32)fsbk.Length);

crinfo.ImageData = bytes;

crinfo.BkImageData = bytesbk;

crinfo.ImageName = filename;

crinfo.ImageType = contenttype;

crinfo.BkImageType = contenttypebk;

}

crinfo.Login = Convert.ToInt32(txtAccno.Value);

crinfo.Name = txtname.Value;

crinfo.CardNo = txtCardNo.Value;

//crinfo.SecurityPin = txtSecuritypin.Value;

crinfo.ExpMonth = ddlMonth.Value;

crinfo.ExpYear = cmbYear.SelectedValue;

crinfo.Type = ddl\_cardType.Value;

int result = 0;

NewAccount\_DataAccess da = new NewAccount\_DataAccess();

DataTable dt = da.getDuplicateAccount(Convert.ToInt32(txtAccno.Value)); // SP\_GetDuplicateAccounts

if (dt.Rows.Count > 0)

{

foreach (DataRow dr in dt.Rows)

{

crinfo.Login = Convert.ToInt32(dr["Login"]);

result = crdata.AddNewCreditCard(crinfo, Convert.ToInt32(Session["UID"])); // SP\_AddNewCard

}

}

else

{

result = crdata.AddNewCreditCard(crinfo, Convert.ToInt32(Session["UID"]));

}

if (result > 0)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Credit Card added Successfully.')", true);

LoadClientDocuments(this, null);

return;

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please check the fields')", true);

return;

}

}

}

catch (Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

Crypto Approval

Load Button click:

protected void LoadClientDocuments(object sender, EventArgs e)

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

int login = 0;

if (txtAccno.Value != "")

{

login = Convert.ToInt32(txtAccno.Value);

}

if (txtAccno.Value != "")

{

txtCardNo.Value = string.Empty;

txtEditCardNo.Value = string.Empty;

txtEditName.Value = string.Empty;

txtFullName.Value = string.Empty;

txtname.Value = string.Empty;

div\_DocList.InnerHtml = string.Empty;

CreditCard\_DataAccess cr = new CreditCard\_DataAccess();

DataTable dt = cr.GetCryptoCardDetails(login); // SP\_GetCryptoCardDetails

StringBuilder html = new StringBuilder();

if (dt.Rows.Count > 0)

{

txtFullName.Value = dt.Rows[0]["Name"].ToString();

txtname.Value = dt.Rows[0]["Name"].ToString();

if (dt.Rows[0]["Id"].ToString() != "0")

{

html.Append("<table class='table table-striped projects'>");

html.Append("<thead><tr>");

foreach (DataColumn column in dt.Columns)

{

if (column.ColumnName != "Id" && column.ColumnName != "ExpYear" && column.ColumnName != "ExpMonth")

{

html.Append("<th>");

html.Append(column.ColumnName);

html.Append("</th>");

}

}

html.Append("</tr></thead><tbody>");

foreach (DataRow row in dt.Rows)

{

html.Append("<tr>");

foreach (DataColumn column in dt.Columns)

{

if (column.ColumnName != "Id" && column.ColumnName != "ExpYear" && column.ColumnName != "ExpMonth")

{

if (column.ColumnName != "Status")

{

if (column.ColumnName == "imageName")

{

html.Append("<td width='150px'><a>");

html.Append(row[column.ColumnName]);

html.Append("</a></td>");

}

else if (column.ColumnName == "CardNo")

{

html.Append("<td width='100px'><a>");

html.Append(row[column.ColumnName]);

html.Append("</a></td>");

}

else

{

html.Append("<td><a>");

html.Append(row[column.ColumnName]);

html.Append("</a></td>");

}

}

else

{

if (row[column.ColumnName].ToString() == "Pending")

{

html.Append("<td><button type='button' class='btn btn-warning btn-xs'>" + row[column.ColumnName] + "</button></td>");

}

else if (row[column.ColumnName].ToString() == "Rejected")

{

html.Append("<td><button type='button' class='btn btn-danger btn-xs'>" + row[column.ColumnName] + "</button></td>");

}

else

{

html.Append("<td><button type='button' class='btn btn-success btn-xs'>" + row[column.ColumnName] + "</button></td>");

}

}

}

}

string appstr = "document.getElementById(" + "&#039;" + "hdndownloadID" + "&#039;).value=&#039;" + row["Id"].ToString() + "&#039;;document.getElementById(" + "&#039;" + "hdnFourDigit" + "&#039;).value=&#039;" + row["CardNo"].ToString() + "&#039;;document.getElementById(" + "&#039;" + "hdnCardType" + "&#039;).value=&#039;" + row["Type"].ToString() + "&#039;";

string host = HttpContext.Current.Request.Url.Host;

string getimage = "";

string getbackimage = "";

string editclick = "";

getimage = "window.open(&#039;http://185.4.178.133:13956/getCreditCardDetails.aspx?login=" + login.ToString() + "&orderno=" + row["Id"].ToString() + "&side=0&#039;);";

getbackimage = "window.open(&#039;http://185.4.178.133:13956/getCreditCardDetails.aspx?login=" + login.ToString() + "&orderno=" + row["Id"].ToString() + "&side=1&#039;);";

if (row["Status"].ToString() == "Approved")

editclick = "document.getElementById(&#039;txtEditCardNo&#039;).value=&#039;" + row["CardNo"].ToString() + "&#039;; document.getElementById(&#039;txtEditCardNo&#039;).setAttribute(&#039;readonly&#039;,&#039;readonly&#039;); document.getElementById(&#039;txtEditName&#039;).value=&#039;" + row["Name"].ToString() + "&#039;; document.getElementById(&#039;ddlyear&#039;).value=&#039;" + row["ExpYear"].ToString() + "&#039;; document.getElementById(&#039;ddledMonth&#039;).value=&#039;" + row["ExpMonth"].ToString() + "&#039;; document.getElementById(" + "&#039;" + "hdndownloadID" + "&#039;).value=&#039;" + row["Id"].ToString() + "&#039;; document.getElementById(&#039;ddlED\_Type&#039;).value=&#039;" + row["Type"].ToString() + "&#039;";

else

editclick = "document.getElementById(&#039;txtEditCardNo&#039;).value=&#039;" + row["CardNo"].ToString() + "&#039;; document.getElementById(&#039;txtEditCardNo&#039;).removeAttribute(&#039;readonly&#039;); document.getElementById(&#039;txtEditName&#039;).value=&#039;" + row["Name"].ToString() + "&#039;; document.getElementById(&#039;ddlyear&#039;).value=&#039;" + row["ExpYear"].ToString() + "&#039;; document.getElementById(&#039;ddledMonth&#039;).value=&#039;" + row["ExpMonth"].ToString() + "&#039;; document.getElementById(" + "&#039;" + "hdndownloadID" + "&#039;).value=&#039;" + row["Id"].ToString() + "&#039;; document.getElementById(&#039;ddlED\_Type&#039;).value=&#039;" + row["Type"].ToString() + "&#039;";

int orderno = Convert.ToInt32(row["Id"].ToString());

string orderno1 = (Request.Form["hdndownloadID"]);

login = Convert.ToInt32(login.ToString());

html.Append("<td><a onclick='" + getimage + "' class='btn btn-primary btn-xs'><i class='fa fa-arrow-down'></i> View image </a><a data-toggle='modal' data-target='#dv\_approve\_Confirm' onclick='" + appstr + "' class='btn btn-success btn-xs'><i class='fa fa-check-square-o'></i> Approve </a><a data-toggle='modal' data-target='#dv\_revert\_comfirm' onclick='" + appstr + "' class='btn btn-danger btn-xs'><i class='fa fa-undo'></i> Reject </a><a class='btn btn-dark btn-xs' data-toggle='modal' data-target='#dv\_addDetails' onclick='" + editclick + "'><i class='fa fa-pencil'></i> Edit </a></td></tr>");

}

html.Append("</tbody></table>");

div\_DocList.InnerHtml = html.ToString();

}

}

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please Enter Account Number.')", true);

txtAccno.Focus();

return;

}

}

catch (Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

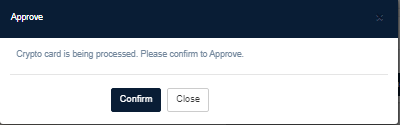
return;

}

}

After listing the cards,

**Approve button click:**



Confirm button click:

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

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

CreditCard\_DataAccess crdata = new CreditCard\_DataAccess();

CreditCard\_Info crinfo = new CreditCard\_Info();

int Result = 0;

int login;

if (txtAccno.Value == "")

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter Login to continue.')", true);

return;

}

SalesLead\_DataAccess salesda = new SalesLead\_DataAccess();

SalesLeadsInfo info1 = new SalesLeadsInfo();

info1 = salesda.GetAcctBasicDts(txtAccno.Value); // SP\_GetAcctBasicDts

login = Convert.ToInt32(txtAccno.Value);

string orderno = (Request.Form["hdndownloadID"]);

string fourdigit = Request.Form["hdnFourDigit"];

string cardtype= Request.Form["hdnCardType"];

if (fourdigit != "" && fourdigit != "0")

{

Result = crdata.UpdateCreditCardStatus(login, Convert.ToInt32(orderno), "Approved", Convert.ToInt32(Session["UID"])); // SP\_VerifyCreditCard

if (Result > 0)

{

EmailFile emf = new EmailFile();

try

{

string em = emf.CreditCardApproved(info1, fourdigit, cardtype);

if (!em.Contains("Email Sent successfully"))

{

System.Threading.Thread.Sleep(10000);

em = emf.CreditCardApproved(info1, fourdigit, cardtype);

em = "[" + em + "]";

}

LoadClientDocuments(this, null);

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Credit Card Updated and Approved.')", true);

return;

}

catch(Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please check the fields')", true);

return;

}

}

}

catch (Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

public string CreditCardApproved(SalesLeadsInfo info, string fourdigit, string cardtype)

{

string res = string.Empty;

try

{

string msg = string.Empty;

string arrtitle = string.Empty;

using (WebClient webClient = new System.Net.WebClient())

{

WebClient objWebClient = new WebClient();

var strcontent = "";

if (cardtype == "Credit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CreditCardApproved.html");

else if (cardtype == "Debit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/DebitCardApproved.html");

else if (cardtype == "Crypto")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CryptoCardApproved.html");

msg = Convert.ToString(strcontent);

}

msg = msg.Replace("[FourDigits]", fourdigit);

if (info.Title != "")

msg = msg.Replace("[NameWithTitle]", info.Title + " " + info.Name);

else

msg = msg.Replace("[NameWithTitle]", info.Name);

string MailServer = "";

string ToEmail = info.Email1;

string ToName = info.Name;

string FromEmail = "";

string FromName = "";

string Subject = "";

if (cardtype == "Credit")

Subject = "Credit Card Approved";

else if (cardtype == "Debit")

Subject = "Debit Card Approved";

else if (cardtype == "Crypto")

Subject = "Crypto Card Approved";

SmtpClient o\_Client;

MailServer = M\_MailServer;

FromEmail = M\_SalesFromEmail;

FromName = M\_FromName;

o\_Client = new SmtpClient(MailServer);

o\_Client.Credentials = new System.Net.NetworkCredential(M\_SalesFromEmail, M\_SalesEmailPassword);

o\_Client.Port = M\_Port;

o\_Client.Timeout = 25000;

o\_Client.EnableSsl = true;

o\_Client.Host = M\_MailServer;

MailMessage o\_Message = new MailMessage();

o\_Message.From = new MailAddress(FromEmail, FromName);

int iname = 0;

foreach (string to in ToEmail.Split(';'))

{

if (ToName.Contains(';'))

o\_Message.To.Add(new MailAddress(to, ToName.Split(';')[iname]));

else

o\_Message.To.Add(new MailAddress(to, ToName));

iname++;

}

o\_Message.Bcc.Add("crm@trusttc.com");

o\_Message.Bcc.Add("magt@trusttc.com");

o\_Message.Bcc.Add("backoffice@trusttc.com");

o\_Message.Subject = Subject;

o\_Message.IsBodyHtml = true;

o\_Message.Body = msg;

System.Net.ServicePointManager.ServerCertificateValidationCallback = delegate (object s,

System.Security.Cryptography.X509Certificates.X509Certificate certificate,

System.Security.Cryptography.X509Certificates.X509Chain chain,

System.Net.Security.SslPolicyErrors sslPolicyErrors)

{

return true;

};

o\_Client.Send(o\_Message);

o\_Message.Dispose();

res = "Email Sent successfully";

}

catch (Exception ex)

{

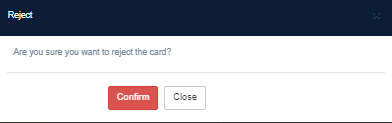
res = ex.Message;

}

return res;

}

Reject button Click:



Confirm click:

public void Revert\_Click(object sender, EventArgs e)

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

SalesLead\_DataAccess salesda = new SalesLead\_DataAccess();

SalesLeadsInfo info1 = new SalesLeadsInfo();

info1 = salesda.GetAcctBasicDts(txtAccno.Value); // SP\_GetAcctBasicDts

int orderno = 0;

int login = 0;

int result = 0;

if (txtAccno.Value != "")

{

login = Convert.ToInt32(txtAccno.Value);

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter Login to continue.')", true);

return;

}

string fourdigit = Request.Form["hdnFourDigit"];

orderno = Convert.ToInt32(Request.Form["hdndownloadID"]);

string cardtype = Request.Form["hdnCardType"];

result = cr.UpdateCreditCardStatus(login, orderno, "Rejected", Convert.ToInt32(Session["UID"])); // SP\_VerifyCreditCard

EmailFile emf = new EmailFile();

string em = emf.CreditCardRejected(info1, fourdigit, cardtype);

LoadClientDocuments(this, null);

}

catch (Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

public string CreditCardRejected(SalesLeadsInfo info, string fourdigits, string cardtype)

{

string res = string.Empty;

try

{

string msg = string.Empty;

string arrtitle = string.Empty;

using (WebClient webClient = new System.Net.WebClient())

{

WebClient objWebClient = new WebClient();

var strcontent = "";

if (cardtype == "Credit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CreditCardRejected.html");

else if (cardtype == "Debit")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/DebitCardRejected.html");

else if (cardtype == "Crypto")

strcontent = objWebClient.DownloadString("http://localhost:13956/EmailTemplates/EN/CryptoCardRejected.html");

msg = Convert.ToString(strcontent);

}

msg = msg.Replace("[FourDigits]", fourdigits.ToString());

if (info.Title != "")

msg = msg.Replace("[NameWithTitle]", info.Title + " " + info.Name);

else

msg = msg.Replace("[NameWithTitle]", info.Name);

string MailServer = "";

string ToEmail = info.Email1;

string ToName = info.Name;

string FromEmail = "";

string FromName = "";

string Subject = "";

if (cardtype == "Credit")

Subject = "Credit Card Rejected";

else if (cardtype == "Debit")

Subject = "Debit Card Rejected";

else if (cardtype == "Crypto")

Subject = "Crypto Card Rejected";

SmtpClient o\_Client;

MailServer = M\_MailServer;

FromEmail = M\_TransactionsFromEmail;

FromName = M\_FromName;

o\_Client = new SmtpClient(MailServer);

o\_Client.Credentials = new System.Net.NetworkCredential(M\_TransactionsFromEmail, M\_TransactionsEmailPassword);

o\_Client.Port = M\_Port;

o\_Client.Timeout = 25000;

o\_Client.EnableSsl = true;

o\_Client.DeliveryMethod = SmtpDeliveryMethod.Network;

o\_Client.Host = M\_MailServer;

MailMessage o\_Message = new MailMessage();

o\_Message.From = new MailAddress(FromEmail, FromName);

int iname = 0;

foreach (string to in ToEmail.Split(';'))

{

if (ToName.Contains(';'))

o\_Message.To.Add(new MailAddress(to, ToName.Split(';')[iname]));

else

o\_Message.To.Add(new MailAddress(to, ToName));

iname++;

}

o\_Message.Bcc.Add("crm@trusttc.com");

o\_Message.Bcc.Add("magt@trusttc.com");

o\_Message.Bcc.Add("backoffice@trusttc.com");

o\_Message.Subject = Subject;

o\_Message.IsBodyHtml = true;

o\_Message.Body = msg;

System.Net.ServicePointManager.ServerCertificateValidationCallback = delegate (object s,

System.Security.Cryptography.X509Certificates.X509Certificate certificate,

System.Security.Cryptography.X509Certificates.X509Chain chain,

System.Net.Security.SslPolicyErrors sslPolicyErrors)

{

return true;

};

o\_Client.Send(o\_Message);

o\_Message.Dispose();

res = "Email Sent successfully";

}

catch (Exception ex)

{

res = ex.Message;

}

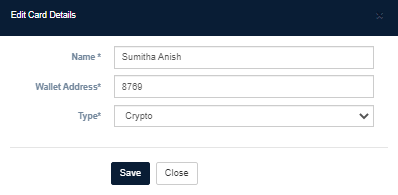
return res;

}

**View Image:**

Display the image.

**Edit Card Details:**



Save click : Use the same function btnUpdate\_Click

Add Wallet Click:

$("#btnaddCard").click(function () {

if ($('#txtAccno').val() == "") {

alert('Please enter the Account No. to continue.');

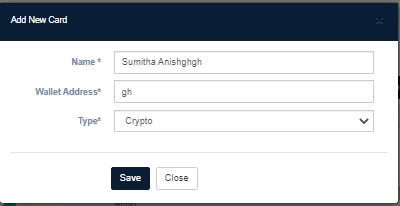
return;

}

$("#btnaddCard").attr("data-toggle", "modal");

$("#btnaddCard").attr("data-target", "#dv\_AddCard");

});



Add the option to upload image also same like credit card.

Save button click:

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

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

CreditCard\_DataAccess crdata = new CreditCard\_DataAccess();

CreditCard\_Info crinfo = new CreditCard\_Info();

int Result = 0;

int login;

string orderno = "0";

string approve = Request.Form["hdnApprove"];

login = Convert.ToInt32(txtAccno.Value);

if (Request.Form["hdndownloadID"] != null && Request.Form["hdndownloadID"] != "")

orderno = (Request.Form["hdndownloadID"]);

else

orderno = "0";

if (txtEditName.Value == "")

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter the Name')", true);

return;

}

if (txtEditCardNo.Value == "" || txtEditCardNo.Value == "0" || txtEditCardNo.Value.Length < 4 || Convert.ToInt32(txtEditCardNo.Value) == 0)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter the wallet address')", true);

return;

}

crinfo.Name = txtEditName.Value;

crinfo.CardNo = txtEditCardNo.Value;

crinfo.SecurityPin = "";

crinfo.ExpMonth = "";// ddledMonth.Value;

crinfo.ExpYear = "";// ddlyear.SelectedValue;

crinfo.Type = ddlED\_Type.Value;

Result = crdata.updateCreditCard(crinfo, login, orderno, Convert.ToInt32(Session["UID"])); // SP\_UpdateCreditCardDetails

btnUpload\_Click(orderno);

if (Result > 0)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Credit Card Updated Successfully.')", true);

LoadClientDocuments(this, null);

return;

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please check the fields')", true);

return;

}

}

catch (Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

**Bank Approval**

Load button

protected void LoadClientDocuments(object sender, EventArgs e)

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

int login = 0;

if (txtAccno.Value != "")

{

login = Convert.ToInt32(txtAccno.Value);

}

if (txtAccno.Value != "")

{

txtAccntNo.Value = string.Empty;

txtBankAddr.Value = string.Empty;

txtBankName.Value = string.Empty;

txtFullName.Value = string.Empty;

txtOtherInfo.Value = string.Empty;

txtSwiftCode.Value = string.Empty;

CreditCard\_DataAccess cda = new CreditCard\_DataAccess();

DataTable dt = cda.getBankDetails(login); // SP\_GetBankDetails

StringBuilder html = new StringBuilder();

html.Append("<table class='table table-striped projects'>");

html.Append("<thead><tr>");

foreach (DataColumn column in dt.Columns)

{

if (column.ColumnName != "Id")

{

html.Append("<th>");

html.Append(column.ColumnName);

html.Append("</th>");

}

}

html.Append("</tr></thead><tbody>");

foreach (DataRow row in dt.Rows)

{

html.Append("<tr>");

foreach (DataColumn column in dt.Columns)

{

if (column.ColumnName != "Id")

{

if (column.ColumnName != "Status")

{

html.Append("<td><a>");

html.Append(row[column.ColumnName]);

html.Append("</a></td>");

}

else

{

if (row[column.ColumnName].ToString() == "Pending")

{

html.Append("<td><button type='button' class='btn btn-warning btn-xs'>" + row[column.ColumnName] + "</button></td>");

}

else if (row[column.ColumnName].ToString() == "Rejected")

{

html.Append("<td><button type='button' class='btn btn-danger btn-xs'>" + row[column.ColumnName] + "</button></td>");

}

else

{

html.Append("<td><button type='button' class='btn btn-success btn-xs'>" + row[column.ColumnName] + "</button></td>");

}

}

}

}

string appstr = "document.getElementById(" + "&#039;" + "hdndownloadID" + "&#039;).value=&#039;" + row["Id"].ToString() + "&#039;; document.getElementById(" + "&#039;" + "hdnAccNo" + "&#039;).value=&#039;" + row["Acc\_No"].ToString() + "&#039;";

string editclick = "document.getElementById(&#039;txtFullName&#039;).value=&#039;" + row["Acc\_Name"].ToString() + "&#039;; document.getElementById(&#039;txtAccntNo&#039;).value=&#039;" + row["Acc\_No"].ToString() + "&#039;; document.getElementById(&#039;txtBankName&#039;).value=&#039;" + row["Bank\_Name"].ToString() + "&#039;; document.getElementById(&#039;txtBankAddr&#039;).value=&#039;" + row["Bank\_Addr"].ToString() + "&#039;; document.getElementById(&#039;txtSwiftCode&#039;).value=&#039;" + row["swiftcode"].ToString() + "&#039;; document.getElementById(&#039;txtOtherInfo&#039;).value=&#039;" + row["OtherInfo"].ToString() + "&#039;; document.getElementById(&#039;txtiban&#039;).value=&#039;" + row["iban"].ToString() + "&#039; document.getElementById(" + "&#039;" + "hdndownloadID" + "&#039;).value=&#039;" + row["Id"].ToString() + "&#039;; document.getElementById(" + "&#039;" + "hdnUpdateFlag" + "&#039;).value=&#039;2&#039;";

int orderno = Convert.ToInt32(row["Id"].ToString());

string orderno1 = (Request.Form["hdndownloadID"]);

login = Convert.ToInt32(login.ToString());

html.Append("<td><a data-toggle='modal' data-target='#dv\_approve\_Confirm' onclick='" + appstr + "' class='btn btn-success btn-xs'><i class='fa fa-check-square-o'></i> Approve </a><a data-toggle='modal' data-target='#dv\_revert\_comfirm' onclick='" + appstr + "' class='btn btn-danger btn-xs'><i class='fa fa-undo'></i> Reject </a><a class='btn btn-dark btn-xs' data-toggle='modal' data-target='#dv\_addDetails' onclick='" + editclick + "'><i class='fa fa-pencil'></i> Edit </a></td></tr>");

}

html.Append("</tbody></table>");

div\_DocList.InnerHtml = html.ToString();

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please Enter Account Number.')", true);

txtAccno.Focus();

return;

}

}

catch (Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

Save Bank Details

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

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

if (txtAccno.Value == "" || txtAccno.Value == "0")

{

txtAccno.Focus();

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please Enter Login.')", true);

return;

}

if (txtAccntNo.Value == "" || txtAccntNo.Value == "0")

{

txtAccntNo.Focus();

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please Enter Account Number.')", true);

return;

}

if (txtBankName.Value == "" || txtBankName.Value == "0")

{

txtBankName.Focus();

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please Enter Bank Name.')", true);

return;

}

CreditCard\_DataAccess cda = new CreditCard\_DataAccess();

BankDetailsInfo binfo = new BankDetailsInfo();

binfo.AccName = txtFullName.Value;

binfo.AccNo = txtAccntNo.Value;

binfo.BankAddr = txtBankAddr.Value;

binfo.BankName = txtBankName.Value;

binfo.IBAN = txtiban.Value;

if (hdnUpdateFlag.Value == string.Empty)

binfo.Flag = 1;

else

binfo.Flag = 2;

binfo.Login = Convert.ToInt32(txtAccno.Value);

binfo.OtherInfo = txtOtherInfo.Value;

binfo.SwiftCode = txtSwiftCode.Value;

if (hdndownloadID.Value != "")

binfo.Id = Convert.ToInt32(hdndownloadID.Value);

int res = 0;

NewAccount\_DataAccess da = new NewAccount\_DataAccess();

DataTable dt = da.getDuplicateAccount(Convert.ToInt32(txtAccno.Value)); // SP\_GetDuplicateAccounts

res = cda.SaveBankDetails(binfo, Convert.ToInt32(Session["UID"]));

// SP\_SaveBankDetails

if(res>0)

{

hdnUpdateFlag.Value = string.Empty;

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Bank Details Saved Successfully.')", true);

}

LoadClientDocuments(null, null);

}

catch(Exception ex)

{

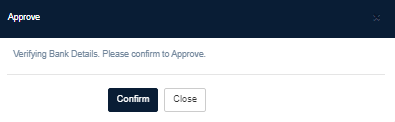
ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

Approve button click



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

{

try

{

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

{

Response.Redirect("Login.aspx");

return;

}

CreditCard\_DataAccess crdata = new CreditCard\_DataAccess();

BankDetailsInfo binfo = new BankDetailsInfo();

int Result = 0;

int login;

if (txtAccno.Value == "")

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter Login to continue.')", true);

return;

}

login = Convert.ToInt32(txtAccno.Value);

string orderno = (Request.Form["hdndownloadID"]);

string accno = Request.Form["hdnAccNo"];

Result = crdata.UpdateDocStatus(login, Convert.ToInt32(orderno), "Approved", Convert.ToInt32(Session["UID"])); // SP\_VerifyBankDetails

if (Result > 0)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Bank details Approved successfully.')", true);

LoadClientDocuments(this, null);

return;

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please check the fields')", true);

return;

}

}

catch (Exception ex)

{

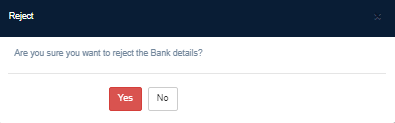
ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}

Reject click



public void Revert\_Click(object sender, EventArgs e)

{

try

{

CreditCard\_DataAccess crdata = new CreditCard\_DataAccess();

BankDetailsInfo binfo = new BankDetailsInfo();

int orderno = 0;

int login = 0;

int result = 0;

if (txtAccno.Value != "")

{

login = Convert.ToInt32(txtAccno.Value);

}

else

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('Please enter Login to continue.')", true);

return;

}

string accno = Request.Form["hdnAccNo"];

orderno = Convert.ToInt32(Request.Form["hdndownloadID"]);

result = crdata.UpdateDocStatus(login, orderno, "Rejected", Convert.ToInt32(Session["UID"])); // SP\_VerifyBankDetails

LoadClientDocuments(this, null);

}

catch (Exception ex)

{

ScriptManager.RegisterClientScriptBlock(this, this.Page.GetType(), "Alert", "alert('" + ex.Message + "')", true);

return;

}

}