using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using iTextSharp;

using iTextSharp.text;

using iTextSharp.text.pdf;

using SMARTEXPK.CommUtlService;

using SMARTEXPK.AppClasses;

namespace SMARTEXPK.ReportForms

{

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

{

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

{

try

{

if (!IsPostBack)

{

DataTable data = (DataTable)Session["TransDetails"];

Document document = new iTextSharp.text.Document(PageSize.A4, 30f, 30f, 30f, 30f);

HttpContext.Current.Response.ContentType = "application/pdf";

HttpContext.Current.Response.AddHeader("content-disposition", "attachment;filename=Transaction\_" + DateTime.Today + ".pdf");

HttpContext.Current.Response.Cache.SetCacheability(HttpCacheability.NoCache);

PdfWriter writer = PdfWriter.GetInstance(document, Response.OutputStream);

document.Open();

PdfContentByte cb = writer.DirectContent;

PdfPTable Salary\_details = new PdfPTable(1);

Salary\_details.DefaultCell.Padding = 4;

Salary\_details.SpacingBefore = 20;

Phrase phrase = new iTextSharp.text.Phrase();

phrase.Add(new Chunk("OUTWARD REMITTANCE RECEIPT", iTextSharp.text.FontFactory.GetFont("TIMES\_ROMAN", 11, Font.BOLD)));

PdfPCell cellss = new PdfPCell(phrase);

cellss.HorizontalAlignment = Element.ALIGN\_CENTER;

cellss.Border = 0;

Salary\_details.AddCell(cellss);

document.Add(Salary\_details);

PdfPTable OT1 = new PdfPTable(2);

OT1.DefaultCell.Padding = 4;

OT1.SpacingBefore = 25;

OT1.WidthPercentage = 93;

float[] widthsNew = new float[] { 80f, 90f };

float[] widths = new float[] { 300f, 90f };

OT1.SetWidths(widthsNew);

PdfPCell Serial\_No = new PdfPCell(new Phrase("Transaction Date : " + data.Rows[0]["RM\_VALUEDATE"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

Serial\_No.HorizontalAlignment = Element.ALIGN\_LEFT;

Serial\_No.Border = 0;

OT1.AddCell(Serial\_No);

PdfPCell TxnRef = new PdfPCell(new Phrase("Txn.Ref. : " + data.Rows[0]["RM\_TRANSREFNO"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

TxnRef.HorizontalAlignment = Element.ALIGN\_LEFT;

TxnRef.Border = 0;

OT1.AddCell(TxnRef);

PdfPCell BranchName = new PdfPCell(new Phrase("Branch Name : " + Session["BrName"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

BranchName.HorizontalAlignment = Element.ALIGN\_LEFT;

BranchName.Border = 0;

OT1.AddCell(BranchName);

PdfPCell Pin = new PdfPCell(new Phrase("Pin : "+ Session[ConstantHelper.BrCode].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

Pin.HorizontalAlignment = Element.ALIGN\_LEFT;

Pin.Border = 0;

OT1.AddCell(Pin);

document.Add(OT1);

PdfPTable RemitterDetails = new PdfPTable(1);

RemitterDetails.SpacingBefore = 10;

RemitterDetails.WidthPercentage = 92;

float[] widthsRemitter = new float[] { 300f };

RemitterDetails.SetWidths(widthsRemitter);

PdfPCell Remitter = new PdfPCell(new Phrase("Remitter Details", new Font(Font.HELVETICA, 8, Font.NORMAL)));

Remitter.Padding = 5;

Remitter.Border = 0;

Remitter.BackgroundColor = Color.LIGHT\_GRAY;

Remitter.HorizontalAlignment = Element.ALIGN\_LEFT;

RemitterDetails.AddCell(Remitter);

document.Add(RemitterDetails);

PdfPTable RemitterDetls = new PdfPTable(3);

RemitterDetls.SpacingBefore = 5;

RemitterDetls.SpacingAfter = 0;

RemitterDetls.WidthPercentage = 92;

float[] widthsRemitterDetls = new float[] { 100f, 100f, 100f };

RemitterDetls.SetWidths(widthsRemitterDetls);

PdfPCell RemitterId = new PdfPCell(new Phrase("Remitter ID :" + data.Rows[0]["RM\_REMID"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

RemitterId.Padding = 3;

RemitterId.HorizontalAlignment = Element.ALIGN\_LEFT;

RemitterId.Border = Rectangle.TOP\_BORDER | Rectangle.LEFT\_BORDER;

RemitterDetls.AddCell(RemitterId);

PdfPCell remitterCatg = new PdfPCell(new Phrase("Remitter Category :" + data.Rows[0]["RM\_CATEGORY"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

remitterCatg.Padding = 3;

remitterCatg.HorizontalAlignment = Element.ALIGN\_LEFT;

remitterCatg.Border = Rectangle.TOP\_BORDER;

RemitterDetls.AddCell(remitterCatg);

PdfPCell remitterType = new PdfPCell(new Phrase("Remitter Type :" + data.Rows[0]["RM\_TYPE"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

remitterType.Padding = 3;

remitterType.HorizontalAlignment = Element.ALIGN\_LEFT;

remitterType.Border = Rectangle.TOP\_BORDER | Rectangle.RIGHT\_BORDER;

RemitterDetls.AddCell(remitterType);

PdfPCell firstName = new PdfPCell(new Phrase("First Name :" + data.Rows[0]["firstname"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

firstName.Padding = 3;

firstName.HorizontalAlignment = Element.ALIGN\_LEFT;

firstName.Border = Rectangle.LEFT\_BORDER;

RemitterDetls.AddCell(firstName);

PdfPCell MiddleName = new PdfPCell(new Phrase("Middle Name :" + data.Rows[0]["MiddleName"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

MiddleName.Padding = 3;

MiddleName.HorizontalAlignment = Element.ALIGN\_LEFT;

MiddleName.Border = 0;

RemitterDetls.AddCell(MiddleName);

PdfPCell LastName = new PdfPCell(new Phrase("Last Name : " + data.Rows[0]["LastName"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

LastName.Padding = 3;

LastName.HorizontalAlignment = Element.ALIGN\_LEFT;

LastName.Border = Rectangle.RIGHT\_BORDER;

RemitterDetls.AddCell(LastName);

PdfPCell Nationality = new PdfPCell(new Phrase("Nationality :" + data.Rows[0]["RM\_NATIONALITY"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

Nationality.Padding = 3;

Nationality.HorizontalAlignment = Element.ALIGN\_LEFT;

Nationality.Border = Rectangle.LEFT\_BORDER;

RemitterDetls.AddCell(Nationality);

PdfPCell IDType = new PdfPCell(new Phrase("ID Type :" + data.Rows[0]["RM\_IDTYPE"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

IDType.Padding = 3;

IDType.HorizontalAlignment = Element.ALIGN\_LEFT;

IDType.Border = 0;

RemitterDetls.AddCell(IDType);

PdfPCell IDNo = new PdfPCell(new Phrase("ID No :" + data.Rows[0]["RM\_IDNUMBER"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

IDNo.Padding = 3;

IDNo.HorizontalAlignment = Element.ALIGN\_LEFT;

IDNo.Border = Rectangle.RIGHT\_BORDER;

RemitterDetls.AddCell(IDNo);

PdfPCell Expiry = new PdfPCell(new Phrase("ID Expiry Date :" + data.Rows[0]["RM\_EXPDATE"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

Expiry.Padding = 3;

Expiry.HorizontalAlignment = Element.ALIGN\_LEFT;

Expiry.Border = Rectangle.LEFT\_BORDER | Rectangle.BOTTOM\_BORDER;

RemitterDetls.AddCell(Expiry);

PdfPCell Mobile = new PdfPCell(new Phrase("Mobile :" + data.Rows[0]["RM\_MOBILENO"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

Mobile.Padding = 3;

Mobile.HorizontalAlignment = Element.ALIGN\_LEFT;

Mobile.Border = Rectangle.BOTTOM\_BORDER;

RemitterDetls.AddCell(Mobile);

PdfPCell demo = new PdfPCell(new Phrase("", new Font(Font.HELVETICA, 7, Font.NORMAL)));

demo.Padding = 3;

demo.HorizontalAlignment = Element.ALIGN\_LEFT;

demo.Border = Rectangle.BOTTOM\_BORDER | Rectangle.RIGHT\_BORDER;

RemitterDetls.AddCell(demo);

document.Add(RemitterDetls);

PdfPTable BeneTransdetails = new PdfPTable(2);

BeneTransdetails.SpacingBefore = 5;

BeneTransdetails.SpacingAfter = 5;

BeneTransdetails.WidthPercentage = 92;

float[] widthsbenehead = new float[] { 150f, 150f };

BeneTransdetails.SetWidths(widthsbenehead);

PdfPCell Beneficiary = new PdfPCell(new Phrase("Beneficiary Details", new Font(Font.HELVETICA, 8, Font.NORMAL)));

Beneficiary.Padding = 5;

Beneficiary.Border = 0;

Beneficiary.BackgroundColor = Color.LIGHT\_GRAY;

Beneficiary.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails.AddCell(Beneficiary);

PdfPCell Transaction = new PdfPCell(new Phrase("Transaction Details", new Font(Font.HELVETICA, 8, Font.NORMAL)));

Transaction.Padding = 5;

Transaction.Border = 0;

Transaction.BackgroundColor = Color.LIGHT\_GRAY;

Transaction.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails.AddCell(Transaction);

document.Add(BeneTransdetails);

PdfPTable BeneTransdetails1 = new PdfPTable(4);

BeneTransdetails1.WidthPercentage = 92;

float[] widthsbene = new float[] { 100f, 100f, 100f, 100f };

BeneTransdetails1.SetWidths(widthsbene);

PdfPCell beneName = new PdfPCell(new Phrase("Name", new Font(Font.HELVETICA, 7, Font.NORMAL)));

beneName.Padding = 5;

beneName.Border = Rectangle.LEFT\_BORDER | Rectangle.TOP\_BORDER;

beneName.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(beneName);

PdfPCell benenameVal = new PdfPCell(new Phrase(data.Rows[0]["BNAME"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

benenameVal.Padding = 5;

benenameVal.Border = Rectangle.RIGHT\_BORDER | Rectangle.TOP\_BORDER;

benenameVal.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(benenameVal);

PdfPCell TransCurrency = new PdfPCell(new Phrase("Currency", new Font(Font.HELVETICA, 7, Font.NORMAL)));

TransCurrency.Padding = 5;

TransCurrency.Border = Rectangle.TOP\_BORDER;

TransCurrency.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(TransCurrency);

PdfPCell currencyVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_PRDCURRENCY"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

currencyVal.Padding = 5;

currencyVal.Border = Rectangle.RIGHT\_BORDER | Rectangle.TOP\_BORDER;

currencyVal.HorizontalAlignment = Element.ALIGN\_RIGHT;

BeneTransdetails1.AddCell(currencyVal);

PdfPCell Destination = new PdfPCell(new Phrase("Destination", new Font(Font.HELVETICA, 7, Font.NORMAL)));

Destination.Padding = 5;

Destination.Border = Rectangle.LEFT\_BORDER;

Destination.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(Destination);

PdfPCell DestinationVal = new PdfPCell(new Phrase(data.Rows[0]["Desination"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

DestinationVal.Padding = 5;

DestinationVal.Border = Rectangle.RIGHT\_BORDER;

DestinationVal.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(DestinationVal);

PdfPCell FcAnount = new PdfPCell(new Phrase("FC Amount", new Font(Font.HELVETICA, 7, Font.NORMAL)));

FcAnount.Padding = 5;

FcAnount.Border = 0;

FcAnount.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(FcAnount);

PdfPCell FcAnountVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_FCAMT"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

FcAnountVal.Padding = 5;

FcAnountVal.Border = Rectangle.RIGHT\_BORDER;

FcAnountVal.HorizontalAlignment = Element.ALIGN\_RIGHT;

BeneTransdetails1.AddCell(FcAnountVal);

PdfPCell Product = new PdfPCell(new Phrase("Product", new Font(Font.HELVETICA, 7, Font.NORMAL)));

Product.Padding = 5;

Product.Border = Rectangle.LEFT\_BORDER;

Product.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(Product);

PdfPCell ProductVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_PRODUCT"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

ProductVal.Padding = 5;

ProductVal.Border = Rectangle.RIGHT\_BORDER;

ProductVal.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(ProductVal);

PdfPCell rate = new PdfPCell(new Phrase("Rate", new Font(Font.HELVETICA, 7, Font.NORMAL)));

rate.Padding = 5;

rate.Border = 0;

rate.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(rate);

PdfPCell rateVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_BCRATE"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

rateVal.Padding = 5;

rateVal.Border = Rectangle.RIGHT\_BORDER;

rateVal.HorizontalAlignment = Element.ALIGN\_RIGHT;

BeneTransdetails1.AddCell(rateVal);

PdfPCell Bank = new PdfPCell(new Phrase("Bank", new Font(Font.HELVETICA, 7, Font.NORMAL)));

Bank.Padding = 5;

Bank.Border = Rectangle.LEFT\_BORDER;

Bank.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(Bank);

PdfPCell Bankval = new PdfPCell(new Phrase(data.Rows[0]["RM\_BENBANK"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

Bankval.Padding = 5;

Bankval.Border = Rectangle.RIGHT\_BORDER;

Bankval.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(Bankval);

PdfPCell Charge = new PdfPCell(new Phrase("Charge", new Font(Font.HELVETICA, 7, Font.NORMAL)));

Charge.Padding = 5;

Charge.Border = 0;

Charge.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(Charge);

PdfPCell ChargeVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_CHRG1"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

ChargeVal.Padding = 5;

ChargeVal.Border = Rectangle.RIGHT\_BORDER;

ChargeVal.HorizontalAlignment = Element.ALIGN\_RIGHT;

BeneTransdetails1.AddCell(ChargeVal);

PdfPCell Branch = new PdfPCell(new Phrase("Branch", new Font(Font.HELVETICA, 7, Font.NORMAL)));

Branch.Padding = 5;

Branch.Border = Rectangle.LEFT\_BORDER;

Branch.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(Branch);

PdfPCell Branchval = new PdfPCell(new Phrase(data.Rows[0]["RM\_BENBRANCH"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

Branchval.Padding = 5;

Branchval.Border = Rectangle.RIGHT\_BORDER;

Branchval.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(Branchval);

PdfPCell Discount = new PdfPCell(new Phrase("Discount", new Font(Font.HELVETICA, 7, Font.NORMAL)));

Discount.Padding = 5;

Discount.Border = 0;

Discount.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(Discount);

PdfPCell DiscountVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_DISCOUNT"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

DiscountVal.Padding = 5;

DiscountVal.Border = Rectangle.RIGHT\_BORDER;

DiscountVal.HorizontalAlignment = Element.ALIGN\_RIGHT;

BeneTransdetails1.AddCell(DiscountVal);

PdfPCell purpose = new PdfPCell(new Phrase("Purpose", new Font(Font.HELVETICA, 7, Font.NORMAL)));

purpose.Padding = 5;

purpose.Border = Rectangle.LEFT\_BORDER;

purpose.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(purpose);

PdfPCell purposeVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_POT"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

purposeVal.Padding = 5;

purposeVal.Border = Rectangle.RIGHT\_BORDER;

purposeVal.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(purposeVal);

PdfPCell NetAmount = new PdfPCell(new Phrase("Net Amount", new Font(Font.HELVETICA, 7, Font.NORMAL)));

NetAmount.Padding = 5;

NetAmount.Border = 0;

NetAmount.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(NetAmount);

PdfPCell NetAmountVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_NETAMOUNT"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

NetAmountVal.Padding = 5;

NetAmountVal.Border = Rectangle.RIGHT\_BORDER;

NetAmountVal.HorizontalAlignment = Element.ALIGN\_RIGHT;

BeneTransdetails1.AddCell(NetAmountVal);

PdfPCell Mobilebene = new PdfPCell(new Phrase("Mobile", new Font(Font.HELVETICA, 7, Font.NORMAL)));

Mobilebene.Padding = 5;

Mobilebene.Border = Rectangle.LEFT\_BORDER | Rectangle.BOTTOM\_BORDER;

Mobilebene.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(Mobilebene);

PdfPCell MobilebeneVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_BENEMOBILE"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

MobilebeneVal.Padding = 5;

MobilebeneVal.Border = Rectangle.RIGHT\_BORDER | Rectangle.BOTTOM\_BORDER;

MobilebeneVal.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(MobilebeneVal);

PdfPCell AmountPaid = new PdfPCell(new Phrase("Amount Paid", new Font(Font.HELVETICA, 7, Font.NORMAL)));

AmountPaid.Padding = 5;

AmountPaid.Border = Rectangle.BOTTOM\_BORDER;

AmountPaid.HorizontalAlignment = Element.ALIGN\_LEFT;

BeneTransdetails1.AddCell(AmountPaid);

PdfPCell AmountPaidVal = new PdfPCell(new Phrase(data.Rows[0]["RM\_AMTRECEIVED"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

AmountPaidVal.Padding = 5;

AmountPaidVal.Border = Rectangle.RIGHT\_BORDER | Rectangle.BOTTOM\_BORDER;

AmountPaidVal.HorizontalAlignment = Element.ALIGN\_RIGHT;

BeneTransdetails1.AddCell(AmountPaidVal);

document.Add(BeneTransdetails1);

PdfPTable TransDetails = new PdfPTable(3);

TransDetails.WidthPercentage = 92;

float[] widthTrans = new float[] { 100f, 100f, 100f };

TransDetails.SetWidths(widthTrans);

PdfPCell TransactionStatus = new PdfPCell(new Phrase("Transaction Status: " + data.Rows[0]["STATUS"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

TransactionStatus.Padding = 5;

TransactionStatus.Border = 0;

TransactionStatus.HorizontalAlignment = Element.ALIGN\_LEFT;

TransDetails.AddCell(TransactionStatus);

PdfPCell PayMode = new PdfPCell(new Phrase("Pay Mode: "+ data.Rows[0]["PAYMODE"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

PayMode.Padding = 5;

PayMode.Border = 0;

PayMode.HorizontalAlignment = Element.ALIGN\_LEFT;

TransDetails.AddCell(PayMode);

PdfPCell demo3 = new PdfPCell(new Phrase("", new Font(Font.HELVETICA, 7, Font.NORMAL)));

demo3.Padding = 5;

demo3.Border = 0;

demo3.HorizontalAlignment = Element.ALIGN\_RIGHT;

TransDetails.AddCell(demo3);

PdfPCell CreatedBy = new PdfPCell(new Phrase("Created By: " + data.Rows[0]["RM\_MADEBY"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

CreatedBy.Padding = 5;

CreatedBy.Border = 0;

CreatedBy.HorizontalAlignment = Element.ALIGN\_LEFT;

TransDetails.AddCell(CreatedBy);

PdfPCell ApprovedBy = new PdfPCell(new Phrase("Approved By: " + data.Rows[0]["RM\_APPBY"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

ApprovedBy.Padding = 5;

ApprovedBy.Border = 0;

ApprovedBy.HorizontalAlignment = Element.ALIGN\_LEFT;

TransDetails.AddCell(ApprovedBy);

PdfPCell demo4 = new PdfPCell(new Phrase("", new Font(Font.HELVETICA, 7, Font.NORMAL)));

demo4.Padding = 5;

demo4.Border = 0;

demo4.HorizontalAlignment = Element.ALIGN\_RIGHT;

TransDetails.AddCell(demo4);

PdfPCell Createdon = new PdfPCell(new Phrase("Created On: "+ data.Rows[0]["RM\_MADEON"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

Createdon.Padding = 5;

Createdon.Border = 0;

Createdon.HorizontalAlignment = Element.ALIGN\_LEFT;

TransDetails.AddCell(Createdon);

PdfPCell ApprovedOn = new PdfPCell(new Phrase("Approved On: " + data.Rows[0]["RM\_APPON"].ToString(), new Font(Font.HELVETICA, 7, Font.NORMAL)));

ApprovedOn.Padding = 5;

ApprovedOn.Border = 0;

ApprovedOn.HorizontalAlignment = Element.ALIGN\_LEFT;

TransDetails.AddCell(ApprovedOn);

PdfPCell Sign = new PdfPCell(new Phrase("Cashier Signature", new Font(Font.HELVETICA, 7, Font.NORMAL)));

Sign.Padding = 5;

Sign.Border = 0;

Sign.HorizontalAlignment = Element.ALIGN\_RIGHT;

TransDetails.AddCell(Sign);

document.Add(TransDetails);

PdfPTable agreementdetails = new PdfPTable(1);

agreementdetails.WidthPercentage = 92;

float[] widthagreement = new float[] { 300f };

agreementdetails.SetWidths(widthagreement);

PdfPCell Agreement = new PdfPCell(new Phrase("i confirm the accuracy of the above details and authorize ANB to send the remittance and I bear any expenses incurred. I also declare thet i have read and realized the terms and conditions stated in the remittance agreement " + data.Rows[0]["RM\_REMID"].ToString() +" (remitter ID). Signed by me and ANB: which in the remittance agreement is considered as non-partial of the receipt.", new Font(Font.HELVETICA, 5, Font.NORMAL)));

Agreement.Padding = 5;

Agreement.Border = 0;

Agreement.HorizontalAlignment = Element.ALIGN\_LEFT;

agreementdetails.AddCell(Agreement);

document.Add(agreementdetails);

PdfPTable custsign = new PdfPTable(2);

custsign.SpacingBefore = 5;

custsign.WidthPercentage = 92;

float[] widthsign = new float[] { 100f, 200f };

custsign.SetWidths(widthsign);

PdfPCell custsignature = new PdfPCell(new Phrase("Customer Signature", new Font(Font.HELVETICA, 8, Font.NORMAL)));

custsignature.Padding = 5;

custsignature.Border = 0;

custsignature.HorizontalAlignment = Element.ALIGN\_LEFT;

custsign.AddCell(custsignature);

PdfPCell custsignatureBox = new PdfPCell(new Phrase("", new Font(Font.HELVETICA, 9, Font.NORMAL)));

custsignatureBox.Padding = 5;

Agreement.Border = 1;

custsignatureBox.HorizontalAlignment = Element.ALIGN\_LEFT;

custsign.AddCell(custsignatureBox);

document.Add(custsign);

PdfPTable violation = new PdfPTable(1);

violation.SpacingBefore = 5;

violation.WidthPercentage = 92;

violation.SetWidths(widthagreement);

PdfPCell violationdetails = new PdfPCell(new Phrase("The information in this report is confidential - Private. If you are not authorized to access this report please do not view the information in it. Any violation thereon may be subject to disciplinary actiob and/or legal prosecution.", new Font(Font.HELVETICA, 5, Font.NORMAL)));

violationdetails.Padding = 5;

violationdetails.Border = 0;

violationdetails.HorizontalAlignment = Element.ALIGN\_LEFT;

violation.AddCell(violationdetails);

document.Add(violation);

PdfPTable printdetails = new PdfPTable(2);

printdetails.SpacingBefore = 20;

printdetails.WidthPercentage = 92;

float[] widthprint = new float[] { 300f, 90f };

printdetails.SetWidths(widthprint);

PdfPCell print = new PdfPCell(new Phrase("Printed On: " + DateTime.Now.ToShortDateString(), new Font(Font.HELVETICA, 5, Font.NORMAL)));

print.Padding = 5;

print.Border = 0;

print.HorizontalAlignment = Element.ALIGN\_LEFT;

printdetails.AddCell(print);

PdfPCell printby = new PdfPCell(new Phrase("Printed By: " + Session["UName"].ToString(), new Font(Font.HELVETICA, 5, Font.NORMAL)));

printby.Padding = 5;

printby.Border = 0;

printby.HorizontalAlignment = Element.ALIGN\_LEFT;

printdetails.AddCell(printby);

document.Add(printdetails);

var rect = new iTextSharp.text.Rectangle(100, 100, 100, 100);

rect.Border = iTextSharp.text.Rectangle.BOX;

rect.BorderColor = new Color(0, 0, 0);

rect.BorderWidth = 1;

cb.Rectangle(rect);

document.Add(rect);

document.Close();

HttpContext.Current.Response.Write(document);

HttpContext.Current.Response.End();

}

}

catch(Exception ex)

{

}

}

}

}

using iTextSharp.text;

using iTextSharp.text.pdf;

using Microsoft.Reporting.WebForms;

using SMARTEXPK.AppClasses;

using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Drawing.Printing;

using System.IO;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace SMARTEXPK.UserControl

{

public partial class Report5 : System.Web.UI.Page

{

private CommonClass objComm = new CommonClass();

Hashtable hshTable = new Hashtable();

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

{

try

{

if (!IsPostBack)

{

if (Session["OutwardRpt"] == "YES")

{

Session["OutwardRpt"] = "NO";

BindReportOutward1();

if (Session["Arabic"] == "YES")

{

Session["Arabic"] = "NO";

BindReportOutward2();

}

}

else

{

Session["OutwardRpt"] = "NO";

BindReport();

}

Session["OutwardRpt"] = "NO";

imgPrint\_Click(imgPrint, null);

}

}

catch (Exception ex)

{

// ApplicationLog.ErrorLog(ex, ConstantHelper.Report + ConstantHelper.Page\_Load);

}

finally

{

GC.Collect();

}

}

private void BindReportOutward1()

{

try

{

DataSet ds = new DataSet();

string Path = string.Empty;

string DataSetName = string.Empty;

int dataexists;

int pathexists;

ds = (DataSet)Session["ReportDataSet5"];

if (ds.Tables.Count > 0)

dataexists = 1;

else

dataexists = 0;

Path = Session["ReportPath5"].ToString();

if (Path != string.Empty)

pathexists = 1;

else

pathexists = 0;

int dsnameexists;

DataSetName = Session["DataSetName5"].ToString();

if (DataSetName != null)

dsnameexists = 1;

else

dsnameexists = 0;

if (dataexists == 1 && pathexists == 1 && dsnameexists == 1)

{

ReportViewer1.LocalReport.EnableExternalImages = true;

int i = -1;

string dsName = DataSetName;

string strRptPath = Server.MapPath(Path);

if (ds.Tables[0].Rows.Count > 0)

{

ReportViewer1.LocalReport.ReportPath = strRptPath;

this.ReportViewer1.PageCountMode = PageCountMode.Actual;

if (ds != null)

{

if (ds.Tables.Count == 1)

{

DataTable dt = ds.Tables[0];

ReportDataSource rd = new ReportDataSource(dsName, dt);

if (Session["RptFormat"] != null && Session["RptFormat"].ToString() == "VOU")

{

System.Drawing.Printing.PageSettings pg = new System.Drawing.Printing.PageSettings();

System.Drawing.Printing.PaperSize size = new PaperSize();

size.RawKind = (int)PaperKind.A5;

pg.Landscape = false;

pg.PaperSize = size;

ReportViewer1.SetPageSettings(pg);

}

ReportViewer1.LocalReport.DataSources.Clear();

ReportViewer1.LocalReport.DataSources.Add(rd);

ReportViewer1.LocalReport.Refresh();

}

else

{

ReportViewer1.LocalReport.DataSources.Clear();

for (int j = 0; j < ds.Tables.Count; j++)

{

DataTable dt = ds.Tables[j];

ReportDataSource rd = new ReportDataSource(dt.TableName, dt);

if (Session["RptFormat"] != null && Session["RptFormat"].ToString() == "VOU")

{

System.Drawing.Printing.PageSettings pg = new System.Drawing.Printing.PageSettings();

System.Drawing.Printing.PaperSize size = new PaperSize();

size.RawKind = (int)PaperKind.A5;

pg.Landscape = false;

pg.PaperSize = size;

ReportViewer1.SetPageSettings(pg);

}

ReportViewer1.LocalReport.DataSources.Clear();

ReportViewer1.LocalReport.DataSources.Add(rd);

ReportViewer1.LocalReport.DataSources.Add(rd);

ReportViewer1.LocalReport.Refresh();

}

}

}

}

hshTable = (Hashtable)Session["Parameters5"];

IDictionaryEnumerator en = hshTable.GetEnumerator();

ReportParameter[] param = new ReportParameter[hshTable.Count];

int count = 0;

while (en.MoveNext())

{

param[count] = new ReportParameter(en.Key.ToString(), en.Value.ToString(), false);

count++;

}

this.ReportViewer1.LocalReport.SetParameters(param);

}

ds.Dispose();

}

catch (Exception ex)

{

// ApplicationLog.ErrorLog(ex, ConstantHelper.Report + ConstantHelper.BindReport);

}

}

private void BindReportOutward2()

{

try

{

DataSet ds = new DataSet();

string Path = string.Empty;

string DataSetName = string.Empty;

int dataexists;

int pathexists;

ds = (DataSet)Session["ReportDataSet5"];

if (ds.Tables.Count > 0)

dataexists = 1;

else

dataexists = 0;

Path = Session["ReportPath5"].ToString();

if (Path != string.Empty)

pathexists = 1;

else

pathexists = 0;

int dsnameexists;

DataSetName = Session["DataSetName5"].ToString();

if (DataSetName != null)

dsnameexists = 1;

else

dsnameexists = 0;

if (dataexists == 1 && pathexists == 1 && dsnameexists == 1)

{

ReportViewer1.LocalReport.EnableExternalImages = true;

int i = -1;

string dsName = DataSetName;

string strRptPath = Server.MapPath(Path);

if (ds.Tables[0].Rows.Count > 0)

{

ReportViewer1.LocalReport.ReportPath = strRptPath;

this.ReportViewer1.PageCountMode = PageCountMode.Actual;

if (ds != null)

{

if (ds.Tables.Count == 1)

{

DataTable dt = ds.Tables[0];

ReportDataSource rd = new ReportDataSource(dsName, dt);

if (Session["RptFormat"] != null && Session["RptFormat"].ToString() == "VOU")

{

System.Drawing.Printing.PageSettings pg = new System.Drawing.Printing.PageSettings();

System.Drawing.Printing.PaperSize size = new PaperSize();

size.RawKind = (int)PaperKind.A5;

pg.Landscape = false;

pg.PaperSize = size;

ReportViewer1.SetPageSettings(pg);

}

ReportViewer1.LocalReport.DataSources.Clear();

ReportViewer1.LocalReport.DataSources.Add(rd);

ReportViewer1.LocalReport.Refresh();

}

else

{

ReportViewer1.LocalReport.DataSources.Clear();

for (int j = 0; j < ds.Tables.Count; j++)

{

DataTable dt = ds.Tables[j];

ReportDataSource rd = new ReportDataSource(dt.TableName, dt);

ReportViewer1.LocalReport.DataSources.Add(rd);

ReportViewer1.LocalReport.Refresh();

}

}

}

}

hshTable = (Hashtable)Session["Parameters5"];

IDictionaryEnumerator en = hshTable.GetEnumerator();

ReportParameter[] param = new ReportParameter[hshTable.Count];

int count = 0;

while (en.MoveNext())

{

param[count] = new ReportParameter(en.Key.ToString(), en.Value.ToString(), false);

count++;

}

this.ReportViewer1.LocalReport.SetParameters(param);

}

ds.Dispose();

}

catch (Exception ex)

{

// ApplicationLog.ErrorLog(ex, ConstantHelper.Report + ConstantHelper.BindReport);

}

}

private void BindReport()

{

try

{

DataSet ds = new DataSet();

string Path = string.Empty;

string DataSetName = string.Empty;

int dataexists;

int pathexists;

ds = (DataSet)Session["ReportDataSet5"];

if (ds.Tables.Count > 0)

dataexists = 1;

else

dataexists = 0;

Path = Session["ReportPath5"].ToString();

if (Path != string.Empty)

pathexists = 1;

else

pathexists = 0;

int dsnameexists;

DataSetName = Session["DataSetName5"].ToString();

if (DataSetName != null)

dsnameexists = 1;

else

dsnameexists = 0;

if (dataexists == 1 && pathexists == 1 && dsnameexists == 1)

{

ReportViewer1.LocalReport.EnableExternalImages = true;

int i = -1;

string dsName = DataSetName;

string strRptPath = Server.MapPath(Path);

if (ds.Tables[0].Rows.Count > 0)

{

ReportViewer1.LocalReport.ReportPath = strRptPath;

this.ReportViewer1.PageCountMode = PageCountMode.Actual;

if (ds != null)

{

if (ds.Tables.Count == 1)

{

DataTable dt = ds.Tables[0];

ReportDataSource rd = new ReportDataSource(dsName, dt);

ReportViewer1.LocalReport.DataSources.Clear();

ReportViewer1.LocalReport.DataSources.Add(rd);

ReportViewer1.LocalReport.Refresh();

}

else

{

ReportViewer1.LocalReport.DataSources.Clear();

for (int j = 0; j < ds.Tables.Count; j++)

{

DataTable dt = ds.Tables[j];

ReportDataSource rd = new ReportDataSource(dt.TableName, dt);

ReportViewer1.LocalReport.DataSources.Add(rd);

ReportViewer1.LocalReport.Refresh();

}

}

}

}

hshTable = (Hashtable)Session["Parameters5"];

IDictionaryEnumerator en = hshTable.GetEnumerator();

ReportParameter[] param = new ReportParameter[hshTable.Count];

int count = 0;

while (en.MoveNext())

{

param[count] = new ReportParameter(en.Key.ToString(), en.Value.ToString(), false);

count++;

}

this.ReportViewer1.LocalReport.SetParameters(param);

}

ds.Dispose();

}

catch (Exception ex)

{

// ApplicationLog.ErrorLog(ex, ConstantHelper.Report + ConstantHelper.BindReport);

}

}

protected void imgPrint\_Click(object sender, ImageClickEventArgs e)

{

try

{

if (Session["OutwardRptDD"] + "".ToString() == "DD")

objComm.Print(ReportViewer1, "A4");

else if (Session["OutwardRptDD"] + "".ToString() == "FC") // For FC Print

objComm.Print(ReportViewer1, "A5", 1); //Parameter value 1 added for storing file in the location

else

objComm.Print(ReportViewer1, "A5");

RemoveSession();

}

catch (Exception ex)

{

ApplicationLog.ErrorLog(ex, ConstantHelper.Report + "imgPrint\_Click()");

RemoveSession();

}

finally

{

//if (ReportViewer1 != null)

//{

//ReportViewer1.LocalReport.Dispose();

// ReportViewer1.Dispose();

GC.Collect();

//}

}

}

protected void RemoveSession()

{

try

{

SessionManager.RemoveItem("OutwardRpt");

SessionManager.RemoveItem("Arabic");

SessionManager.RemoveItem("ReportDataSet");

SessionManager.RemoveItem("ReportPath");

SessionManager.RemoveItem("DataSetName");

SessionManager.RemoveItem("Parameters");

SessionManager.RemoveItem("ReportDataSet5");

SessionManager.RemoveItem("ReportPath5");

SessionManager.RemoveItem("DataSetName5");

SessionManager.RemoveItem("Parameters5");

SessionManager.RemoveItem("OutwardRptDD");

}

catch (Exception ex)

{

ApplicationLog.ErrorLog(ex, ConstantHelper.Report + "RemoveSession");

}

}

protected void ReportViewer1\_Unload(object sender, EventArgs e)

{

try

{

if (ReportViewer1 != null)

{

ReportViewer1.LocalReport.Dispose();

if (ReportViewer1 != null)

{

ReportViewer1.Dispose();

}

}

}

catch (Exception ex)

{

ApplicationLog.ErrorLog(ex, ConstantHelper.Report + "ReportViewer1\_Unload()");

//RemoveSession();

}

}

}

}

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Report5.aspx.cs" Inherits="SMARTEXPK.UserControl.Report5" %>

<!DOCTYPE html>

<%@ Register Assembly="Microsoft.ReportViewer.WebForms, Version=15.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91" Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb" %>

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

<head runat="server">

<title></title>

</head>

<body>

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

<%--<asp:scriptmanager runat="server"></asp:scriptmanager>--%>

<%--<div align="center">

<asp:ImageButton ID="imgPrint" runat="server" ImageUrl="~/Images/print.jpg" OnClientClick="Print();" />

<rsweb:ReportViewer ID="ReportViewer1" runat="server" Width="80%"></rsweb:ReportViewer>--%>

<div>

<asp:ScriptManager runat="server"></asp:ScriptManager>

<asp:ImageButton ID="imgPrint" runat="server" ImageUrl="~/Images/print.jpg" OnClick="imgPrint\_Click" />

<%-- <asp:ImageButton ID="ImageButton1" runat="server" ImageUrl="~/Images/print.jpg" OnClientClick="Print();" /> --%>

<br />

<rsweb:ReportViewer ID="ReportViewer1" OnUnload="ReportViewer1\_Unload" runat="server" Width="100%" Height="600px" EnableTelemetry="false"></rsweb:ReportViewer>

</div>

<script type="text/javascript">

function Print() {

debugger;

var report = document.getElementById("<%=ReportViewer1.ClientID %>");

var div = report.getElementsByTagName("DIV");

var reportContents;

for (var i = 0; i < div.length; i++) {

if (div[i].id.indexOf("VisibleReportContent") != -1) {

reportContents = div[i].innerHTML;

break;

}

}

var frame1 = document.createElement('iframe');

frame1.name = "frame1";

frame1.style.position = "absolute";

//frame1.style.top = "-1000000px";

//frame1.marginHeight="8."

document.body.appendChild(frame1);

var frameDoc = frame1.contentWindow ? frame1.contentWindow : frame1.contentDocument.document ? frame1.contentDocument.document : frame1.contentDocument;

frameDoc.document.open();

frameDoc.document.write('<html><head><title>Report</title>');

frameDoc.document.write('</head><body style = "font-family:Arial;font-size:8pt;>');

//style = "font-family:Times New Roman;font-size:10pt;margin:76px 76px 76px 76px">

frameDoc.document.write(reportContents);

frameDoc.document.write('</body></html>');

frameDoc.document.close();

setTimeout(function () {

window.frames["frame1"].focus();

window.frames["frame1"].print();

document.body.removeChild(frame1);

}, 10);

}

// Print function (require the reportviewer client ID)

</script>

<%--</div>--%>

<asp:HiddenField ID="hfDD" runat="server" />

</form>

</body>

</html>