**1. button Search**

#region "button Search"

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

{

try

{

obj\_receipt.RECEIPT\_NO = txtReceiptNo.Text.Trim();

if (Convert.ToInt32(ddlFillCustomer.SelectedValue) == -1)

{

obj\_receipt.CUSTOMER\_ID = 0;

}

else

{

obj\_receipt.CUSTOMER\_ID = Convert.ToInt32(ddlFillCustomer.SelectedValue);

}

if (Convert.ToInt32(ddlPayment\_Mode.SelectedValue) == -1)

{

obj\_receipt.PAYMENT\_MODE = "";

}

else

{

obj\_receipt.PAYMENT\_MODE = ddlPayment\_Mode.SelectedValue;

}

if (txtFrom\_Date.Text == "")

{

obj\_receipt.FROM\_DATE = "";

}

else

{

DateTime from\_date = DateTime.ParseExact(txtFrom\_Date.Text, "dd-MM-yyyy", CultureInfo.InvariantCulture);

obj\_receipt.FROM\_DATE = from\_date.ToString("yyyy-MM-dd");

}

if (txtTo\_Date.Text == "")

{

obj\_receipt.TO\_DATE = "";

}

else

{

DateTime To\_date = DateTime.ParseExact(txtTo\_Date.Text, "dd-MM-yyyy", CultureInfo.InvariantCulture);

obj\_receipt.TO\_DATE = To\_date.ToString("yyyy-MM-dd");

}

DS = obj\_receipt.SearchReceiptMultiple();

if (DS != null)

{

foreach (DataRow row in DS.Tables[0].Rows)

{

if (TotalAmountR > 0)

{

TotalAmountR = TotalAmountR + Convert.ToDecimal(row["PAYMENT\_RECEIVED"]);

}

else

{

TotalAmountR = Convert.ToDecimal(row["PAYMENT\_RECEIVED"]);

}

}

gvReceipt.DataSource = DS;

gvReceipt.DataBind();

gvReceipt.Dispose();

Panel\_gvReceipt.Visible = true;

btnprint.Visible = true;

Session["SearchReceiptData"] = DS.Tables[0];

}

else

{

lblMsg.Text = "No Record Found";

lblMsg.Visible = true;

Panel\_gvReceipt.Visible = false;

btnprint.Visible = false;

}

}

catch (Exception ex)

{

lblMsg.Text = ex.ToString();

lblMsg.Visible = true;

}

}

#endregion

**2. CLASS FILE**

#region "Search Receipt Multiple"

public DataSet SearchReceiptMultiple()

{

try

{

DS = obj\_cDataAccess.RunSPReturnDataSet("USP\_SEARCH\_RECEIPT\_MULTIPLE"

, new SqlParameter("@FROM\_DATE", FROM\_DATE)

, new SqlParameter("@TO\_DATE", TO\_DATE)

, new SqlParameter("@CUSTOMER\_ID", CUSTOMER\_ID)

, new SqlParameter("@PAYMENT\_MODE", PAYMENT\_MODE)

, new SqlParameter("@RECEIPT\_NO", RECEIPT\_NO)

);

}

catch (Exception)

{ }

return DS;

}

#endregion

**3. STORE PROCEDURE**

USE [KARIYOT]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_SEARCH\_RECEIPT\_MULTIPLE] Script Date: 12/31/2015 12:46:47 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Rakesh Panchal>

-- Create date: <16-MAY-2014>

-- Description: <Description,,>

-- =============================================

ALTER PROCEDURE [dbo].[USP\_SEARCH\_RECEIPT\_MULTIPLE]

@FROM\_DATE DATE=''

,@TO\_DATE DATE=''

,@CUSTOMER\_ID INT=0

,@PAYMENT\_MODE INT=0

,@RECEIPT\_NO VARCHAR(50)=NULL

AS

BEGIN

SET NOCOUNT ON;

SELECT RECEIPT\_NO,CONVERT(varchar(10), PAYMENT\_DATE,105) AS PAYMENT\_DATE,PAYMENT\_RECEIVED,CA.CUSTOMER\_NAME,

CASE

TR.PAYMENT\_MODE

WHEN 1 THEN 'Cheque'

WHEN 2 THEN 'Cash Amount'

WHEN 3 THEN 'Demand Draft'

WHEN 4 THEN 'Bank Transfer'

ELSE

'N/A'

END AS PAYMENT\_MODE

FROM TBL\_Receipt TR

LEFT JOIN TBL\_CUSTOMER\_ACCOUNT CA ON TR.CUSTOMER\_ID=CA.CUSTOMER\_ID

WHERE TR.CUSTOMER\_STATUS='1' AND

RECEIPT\_NO LIKE '%'+@RECEIPT\_NO+'%'

AND

CASE

WHEN ISNULL(@PAYMENT\_MODE,'') = '' THEN 1

WHEN ISNULL(@PAYMENT\_MODE,'') <> '' AND PAYMENT\_MODE = + ISNULL(@PAYMENT\_MODE,'') THEN 1

ELSE 0 END = 1

AND

CASE

WHEN ISNULL(@CUSTOMER\_ID,'') = '' THEN 1

WHEN ISNULL(@CUSTOMER\_ID,'') <> '' AND TR.CUSTOMER\_ID = + ISNULL(@CUSTOMER\_ID,'') THEN 1

ELSE 0 END = 1

AND

CASE

WHEN ISNULL(@FROM\_DATE,'') = '' And ISNULL(@TO\_DATE,'') = '' THEN 1

WHEN ISNULL(@FROM\_DATE,'') = '' And ISNULL(@TO\_DATE,'') <> '' And PAYMENT\_DATE <= @TO\_DATE THEN 1

WHEN ISNULL(@FROM\_DATE,'') <> '' And ISNULL(@TO\_DATE,'') = '' And PAYMENT\_DATE >= @FROM\_DATE THEN 1

WHEN ISNULL(@FROM\_DATE,'') <> '' And ISNULL(@TO\_DATE,'') <> '' And PAYMENT\_DATE between @FROM\_DATE and @TO\_DATE THEN 1

ELSE 0 END = 1

END

**mvc Controllers**

using HotelManagement.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace HotelManagement.Controllers

{

public class EmployeeController : Controller

{

//

// GET: /Employee/

int CompanyId = UserNames.companyId();

int HotelId = UserNames.HotelId();

public ActionResult Index()

{

return View();

}

[HttpGet]

public ActionResult CreateRoles()

{

return View();

}

[HttpPost]

public ActionResult CreateRoles(ExistingRoles ExRoles)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

tbl\_ExistingRoles \_roles = new tbl\_ExistingRoles();

\_roles.CompanyId = CompanyId;

\_roles.HotelId = HotelId;

\_roles.Roles = ExRoles.Role;

HME.tbl\_ExistingRoles.Add(\_roles);

HME.SaveChanges();

return RedirectToAction("ShowRoles");

}

}

public ActionResult DeleteRoles(string checkedIds)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

try

{

var IdsArray = checkedIds.Split('~');

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

{

int p = Convert.ToInt32(IdsArray[i]);

tbl\_ExistingRoles tRtypes = HME.tbl\_ExistingRoles.Where(x => x.Sno == p).FirstOrDefault();

HME.tbl\_ExistingRoles.Remove(tRtypes);

HME.SaveChanges();

}

return Json("Data");

}

catch

{

return Json("Please Delete The Rooms Come Under This Role !");

}

}

}

public JsonResult RoleAvailable(string Role)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

return Json(!HME.tbl\_ExistingRoles.Any(x => x.Roles == Role && x.CompanyId == CompanyId && x.HotelId == HotelId), JsonRequestBehavior.AllowGet);

}

}

public ActionResult ShowRoles()

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

List<tbl\_ExistingRoles> \_exroles = HME.tbl\_ExistingRoles.Where(x => x.CompanyId == CompanyId && x.HotelId == HotelId).ToList();

List<ExistingRoles> \_modexroles = new List<ExistingRoles>();

foreach (var lst in \_exroles)

{

ExistingRoles exroles = new ExistingRoles();

exroles.CompanyId = (int)lst.CompanyId;

exroles.HotelId = (int)lst.HotelId;

exroles.Sno = (int)lst.Sno;

exroles.Role = lst.Roles;

\_modexroles.Add(exroles);

}

return View(\_modexroles);

}

}

[HttpGet]

public ActionResult EditRoles(int id)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

tbl\_ExistingRoles \_exroles = HME.tbl\_ExistingRoles.Where(x => x.Sno == id).FirstOrDefault();

ExistingRoles exroles = new ExistingRoles();

exroles.Sno = \_exroles.Sno;

exroles.Role = \_exroles.Roles;

return View(exroles);

}

}

public ActionResult EditRoles(ExistingRoles exrole)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

tbl\_ExistingRoles \_exroles = HME.tbl\_ExistingRoles.Where(x => x.Sno == exrole.Sno).FirstOrDefault();

\_exroles.Roles = exrole.Role;

HME.SaveChanges();

return RedirectToAction("ShowRoles");

}

}

[HttpGet]

public ActionResult CreateEmployee()

{

using(HotelManagementEntities HME=new HotelManagementEntities())

{

List<int> dbDate = new List<int>();

for (int i = 1; i <= 31; i++)

{

dbDate.Add(i);

}

ViewBag.DobDate = dbDate;

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

dbMonth.Add("January");

dbMonth.Add("February");

dbMonth.Add("March");

dbMonth.Add("April");

dbMonth.Add("May");

dbMonth.Add("June");

dbMonth.Add("July");

dbMonth.Add("August");

dbMonth.Add("September");

dbMonth.Add("October");

dbMonth.Add("November");

dbMonth.Add("December");

ViewBag.DobMonth = dbMonth;

List<int> dbYear = new List<int>();

for (int i = 1960; i <= 2011; i++)

{

dbYear.Add(i);

}

ViewBag.DobYear = dbYear;

ViewBag.Departments = HME.tbl\_Department.Where(x => x.Company\_ID == CompanyId && x.Hotel\_ID == HotelId).ToList();

ViewBag.Roles = HME.tbl\_ExistingRoles.Where(x => x.CompanyId == CompanyId && x.HotelId == HotelId).ToList();

return View();

}

}

[HttpPost]

public ActionResult CreateEmployee(CreateEmployee emp)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

tbl\_EmployeeDetail \_empdetail = new tbl\_EmployeeDetail();

\_empdetail.CompanyId = CompanyId;

\_empdetail.HotelId = HotelId;

\_empdetail.EmployeeId = emp.EmployeeId;

\_empdetail.FirstName = emp.FirstName;

\_empdetail.LastName = emp.LastName;

\_empdetail.Address = emp.Address;

\_empdetail.Country = emp.Country;

\_empdetail.State = emp.State;

\_empdetail.City = emp.City;

\_empdetail.PhoneNo = emp.PhoneNo;

\_empdetail.MobileNo = emp.MobileNo;

\_empdetail.EmailId = emp.EmailId;

\_empdetail.DobDate = emp.DobDate;

\_empdetail.DobMonth = emp.DobMonth;

\_empdetail.DobYear = emp.DobYear;

\_empdetail.Department = emp.Department;

\_empdetail.Designation = emp.Designation;

\_empdetail.Role = emp.Role;

\_empdetail.UserName = emp.UserName;

\_empdetail.Password = emp.Password;

HME.tbl\_EmployeeDetail.Add(\_empdetail);

ViewBag.Msg = "Employee Created Successfully";

HME.SaveChanges();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

List<int> dbDate = new List<int>();

for (int i = 1; i <= 31; i++)

{

dbDate.Add(i);

}

ViewBag.DobDate = dbDate;

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

dbMonth.Add("January");

dbMonth.Add("February");

dbMonth.Add("March");

dbMonth.Add("April");

dbMonth.Add("May");

dbMonth.Add("June");

dbMonth.Add("July");

dbMonth.Add("August");

dbMonth.Add("September");

dbMonth.Add("October");

dbMonth.Add("November");

dbMonth.Add("December");

ViewBag.DobMonth = dbMonth;

List<int> dbYear = new List<int>();

for (int i = 1960; i <= 2011; i++)

{

dbYear.Add(i);

}

ViewBag.DobYear = dbYear;

ViewBag.Departments = HME.tbl\_Department.Where(x => x.Company\_ID == CompanyId && x.Hotel\_ID == HotelId).ToList();

ViewBag.Roles = HME.tbl\_ExistingRoles.Where(x => x.CompanyId == CompanyId && x.HotelId == HotelId).ToList();

return RedirectToAction("ShowEmployees", new {id="1" });

}

}

public ActionResult ShowEmployees(string id)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

List<tbl\_EmployeeDetail> emplist = HME.tbl\_EmployeeDetail.Where(x => x.CompanyId == CompanyId && x.HotelId == HotelId).ToList();

List<CreateEmployee> employ = new List<CreateEmployee>();

foreach (var lst in emplist)

{

CreateEmployee emp = new CreateEmployee();

emp.Sno = lst.Sno;

emp.EmployeeId = lst.EmployeeId;

emp.FirstName = lst.FirstName;

emp.LastName = lst.LastName;

emp.EmailId=lst.EmailId;

emp.Department=(int)lst.Department;

emp.DepartmentName=lst.tbl\_Department.Dept\_Name;

emp.RoleName = lst.tbl\_ExistingRoles.Roles;

emp.Designation = lst.Designation;

employ.Add(emp);

if (id == "1")

{

ViewBag.Msg = "Record Saved Successfully";

}

else if(id=="2")

{

ViewBag.Msg = "Record Updated Successfully";

}

}

return View(employ);

}

}

public ActionResult EditEmployee(int id)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

tbl\_EmployeeDetail \_empdetail = HME.tbl\_EmployeeDetail.Where(x => x.Sno == id).FirstOrDefault();

CreateEmployee cemployee = new CreateEmployee();

cemployee.Sno = \_empdetail.Sno;

cemployee.EmployeeId = \_empdetail.EmployeeId;

cemployee.FirstName = \_empdetail.FirstName;

cemployee.LastName = \_empdetail.LastName;

cemployee.Address = \_empdetail.Address;

cemployee.Country = \_empdetail.Country;

cemployee.State = \_empdetail.State;

cemployee.City = \_empdetail.City;

cemployee.PhoneNo = (Int64)\_empdetail.PhoneNo;

cemployee.MobileNo = (Int64)\_empdetail.MobileNo;

cemployee.EmailId = \_empdetail.EmailId;

cemployee.DobDate = (int)\_empdetail.DobDate;

cemployee.DobMonth = \_empdetail.DobMonth;

cemployee.DobYear = (int)\_empdetail.DobYear;

cemployee.Department = (int)\_empdetail.Department;

cemployee.Designation = \_empdetail.Designation;

cemployee.Role = (int)\_empdetail.Role;

cemployee.UserName = \_empdetail.UserName;

cemployee.Password = \_empdetail.Password;

List<int> dbDate = new List<int>();

for (int i = 1; i <= 31; i++)

{

dbDate.Add(i);

}

ViewBag.DobDate = dbDate;

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

dbMonth.Add("January");

dbMonth.Add("February");

dbMonth.Add("March");

dbMonth.Add("April");

dbMonth.Add("May");

dbMonth.Add("June");

dbMonth.Add("July");

dbMonth.Add("August");

dbMonth.Add("September");

dbMonth.Add("October");

dbMonth.Add("November");

dbMonth.Add("December");

ViewBag.DobMonth = dbMonth;

List<int> dbYear = new List<int>();

for (int i = 1960; i <= 2011; i++)

{

dbYear.Add(i);

}

ViewBag.DobYear = dbYear;

ViewBag.Departments = HME.tbl\_Department.Where(x => x.Company\_ID == CompanyId && x.Hotel\_ID == HotelId).ToList();

ViewBag.Roles = HME.tbl\_ExistingRoles.Where(x => x.CompanyId == CompanyId && x.HotelId == HotelId).ToList();

return View(cemployee);

}

}

[HttpPost]

public ActionResult EditEmployee(CreateEmployee emp)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

tbl\_EmployeeDetail \_empdetail = HME.tbl\_EmployeeDetail.Where(x => x.CompanyId == CompanyId && x.HotelId == HotelId).FirstOrDefault();

\_empdetail.FirstName = emp.FirstName;

\_empdetail.LastName = emp.LastName;

\_empdetail.Address = emp.Address;

\_empdetail.Country = emp.Country;

\_empdetail.State = emp.State;

\_empdetail.City = emp.City;

\_empdetail.PhoneNo = emp.PhoneNo;

\_empdetail.MobileNo = emp.MobileNo;

\_empdetail.DobDate = emp.DobDate;

\_empdetail.DobMonth = emp.DobMonth;

\_empdetail.DobYear = emp.DobYear;

\_empdetail.Department = emp.Department;

\_empdetail.Designation = emp.Designation;

\_empdetail.Role = emp.Role;

\_empdetail.UserName = emp.UserName;

\_empdetail.Password = emp.Password;

HME.SaveChanges();

return RedirectToAction("ShowEmployees", new {id="2" });

}

}

public JsonResult IsEmployeeId(string EmployeeId,string Sno)

{

using (HotelManagementEntities HME = new HotelManagementEntities())

{

int sn = Convert.ToInt32(Sno);

return Json(!HME.tbl\_EmployeeDetail.Any(x => x.EmployeeId == EmployeeId && x.CompanyId == CompanyId && x.HotelId == HotelId && x.Sno!=sn), JsonRequestBehavior.AllowGet);

}

}

public ActionResult Shifts()

{

return View();

}

}

}

**model**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.ComponentModel.DataAnnotations;

namespace HotelManagement.Models

{

public class CreateEmployee

{

public int Sno { get; set; }

public int CompanyId { get; set; }

public int HotelId { get; set; }

[Required(ErrorMessage="Please Enter EmployeeId")]

[Remote("IsEmployeeId", "Employee", ErrorMessage = "EmployeeId Already Exist",AdditionalFields="Sno")]

public string EmployeeId { get; set; }

[Required(ErrorMessage="Please Enter FirstName")]

public string FirstName { get; set; }

[Required(ErrorMessage = "Please Enter LastName")]

public string LastName { get; set; }

[Required(ErrorMessage="Please Enter Address")]

public string Address { get; set; }

[Required(ErrorMessage="Please Select Country")]

public string Country { get; set; }

[Required(ErrorMessage="Please Select State")]

public string State { get; set; }

[Required(ErrorMessage="Please Enter City")]

public string City { get; set; }

[Required(ErrorMessage="Please Enter PhoneNumber")]

public Int64 PhoneNo { get; set; }

[Required(ErrorMessage = "Please Enter MobileNumber")]

public Int64 MobileNo { get; set; }

[Required(ErrorMessage = "Please Enter EmailId")]

public string EmailId { get; set; }

public int DobDate { get; set; }

public string DobMonth { get; set; }

public int DobYear { get; set; }

[Required(ErrorMessage="Please Select Department")]

public int Department { get; set; }

public string DepartmentName { get; set; }

[Required(ErrorMessage="Please Enter Designation")]

public string Designation { get; set; }

[Required(ErrorMessage = "Please Select Role")]

public int Role { get; set; }

public string RoleName { get; set; }

[Required(ErrorMessage="Please Enter UserName")]

public string UserName { get; set; }

[Required(ErrorMessage="Please Enter Password")]

public string Password { get; set; }

}

}