C:\Users\raj27\source\repos\Workforce\Workforce\Views\Home\Contact.cshtml

Controler

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Mail;

using System.Web;

using System.Web.Mvc;

using Workforce.Models;

using System.IO;

namespace Workforce.Controllers

{

public class HomeController : Controller

{

WorkforceEntities db = new WorkforceEntities();

public ActionResult Index()

{

return View();

}

public ActionResult About()

{

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

return View();

}

[HttpGet]

public ActionResult Addemployee()

{

ViewBag.Success = TempData["Success"];

var citylist = new List<SelectListItem> {

new SelectListItem { Text = "Ahmedabad",Value = "Ahmedabad"},

new SelectListItem { Text = "Mumbai",Value = "Mumbai"},

new SelectListItem { Text ="Delhi", Value="Delhi"},

new SelectListItem { Text ="Chennai", Value="Chennai"},

new SelectListItem { Text ="Hyderabad", Value="Hyderabad"},

new SelectListItem { Text ="Banglore", Value="Banglore"},

new SelectListItem { Text ="Pune", Value="Pune"},

};

var experience = new List<SelectListItem> {

new SelectListItem { Text ="0-1 Year", Value="0-1"},

new SelectListItem { Text ="1-3 Year", Value="1-3"},

new SelectListItem { Text ="3-5 Year", Value="3-5"},

new SelectListItem { Text ="5+ Year", Value="5+"}

};

ViewBag.Citylist = citylist;

ViewBag.Experience = experience;

AddemployeeModel aempm = new AddemployeeModel();

return View(aempm);

}

[HttpPost]

public ActionResult Addemployee(AddemployeeModel model)

{

string ResumeName = Path.GetFileNameWithoutExtension(model.ResumeFile.FileName);

string ResumeExtension = Path.GetExtension(model.ResumeFile.FileName);

//Add Current Date and time To Attached File Name

ResumeName = DateTime.Now.ToString("yyyyMMddhhmmss") + "-" + ResumeName.Trim() + ResumeExtension;

string Resumepath = Path.Combine(Server.MapPath("~/Resume"), ResumeName);

model.ResumeFile.SaveAs(Resumepath);

string PhotoName = Path.GetFileNameWithoutExtension(model.PhotoFile.FileName);

string PhotoExtension = Path.GetExtension(model.PhotoFile.FileName);

//Add Current Date and time To Attached File Name

PhotoName = DateTime.Now.ToString("yyyyMMddhhmmss") + "-" + PhotoName.Trim() + PhotoExtension;

string PhotoPath = Path.Combine(Server.MapPath("~/Image"), PhotoName);

model.PhotoFile.SaveAs(PhotoPath);

wf\_emp wfemp = new wf\_emp

{

Photo = PhotoName,

Resume= ResumeName,

Id = model.id,

Employeeid = model.Eemployeeid,

Name = model.Name,

Dob = model.Dob,

Gender = model.Gender,

Address = model.Address,

City = model.City,

Email = model.Email,

Mobile = model.Mobile,

Experience = model.Experience,

Password = model.Password

};

db.wf\_emp.Add(wfemp);

db.SaveChanges();

//using (MailMessage mm = new MailMessage(new MailAddress("Dotnet2784@gmail.com", "Highfive Rewards"), new MailAddress(model.Email)))

// //customise name here

//{

// try

// {

// //customise subject here

// mm.Subject = "Welcome to Highfive Reward!";

// //customise mail body here

// mm.Body = "<a href='http://xyz.com/account/verify/code'>Hello, Thanks for Register with us!!</a>";

// mm.IsBodyHtml = true;

// SmtpClient smtp = new SmtpClient();

// smtp.Host = /\*"smtp-mail.outlook.com";\*/"smtp.gmail.com";

// smtp.EnableSsl = true;

// NetworkCredential NetworkCred = new NetworkCredential("Dotnet2784@gmail.com", "dotnetmvc");

// smtp.UseDefaultCredentials = false; //email & password

// smtp.Credentials = NetworkCred;

// smtp.Port = 587;

// smtp.Send(mm);

// }

// catch (Exception ex)

// {

// throw ex;

// }

//}

TempData["Success"] = "Employee data added successfully.";

return RedirectToAction("Addemployee", "Home");

}

public JsonResult IsEmailAvailable(string Email)

{

return Json (!db.wf\_emp.Any(i => i.Email == Email), JsonRequestBehavior.AllowGet);

}

[HttpGet]

public ActionResult Employeelist()

{

List<AddemployeeModel> employeelist = db.wf\_emp.Select(i => new AddemployeeModel

{

Photo=i.Photo,

Resume = i.Resume,

id = i.Id,

Eemployeeid=i.Employeeid,

Name=i.Name,

Dob=i.Dob,

Gender=i.Gender,

Address=i.Address,

City=i.City,

Email=i.Email,

Mobile=i.Mobile,

Experience=i.Experience,

Password=i.Password,

}).ToList();

return View(employeelist);

}

[HttpGet]

public ActionResult UpdateEmployee(Int32 id)

{

ViewBag.Success = TempData["Success"];

var citylist = new List<SelectListItem> {

new SelectListItem { Text = "Ahmedabad",Value = "Ahmedabad"},

new SelectListItem { Text = "Mumbai",Value = "Mumbai"},

new SelectListItem { Text ="Delhi", Value="Delhi"},

new SelectListItem { Text ="Chennai", Value="Chennai"},

new SelectListItem { Text ="Hyderabad", Value="Hyderabad"},

new SelectListItem { Text ="Banglore", Value="Banglore"},

new SelectListItem { Text ="Pune", Value="Pune"},

};

var experience = new List<SelectListItem> {

new SelectListItem { Text ="0-1 Year", Value="0-1"},

new SelectListItem { Text ="1-3 Year", Value="1-3"},

new SelectListItem { Text ="3-5 Year", Value="3-5"},

new SelectListItem { Text ="5+ Year", Value="5+"}

};

ViewBag.Citylist = citylist;

ViewBag.Experience = experience;

var wfemp = db.wf\_emp.Where(i => i.Id == id).FirstOrDefault();

AddemployeeModel model = new AddemployeeModel();

if (wfemp != null)

{

model.id = wfemp.Id;

model.Eemployeeid = wfemp.Employeeid;

model.Name = wfemp.Name;

model.Dob = wfemp.Dob;

model.Gender = wfemp.Gender;

model.Address = wfemp.Address;

model .City = wfemp.City;

model.Email = wfemp.Email;

model.Mobile = wfemp.Mobile;

model.Experience = wfemp.Experience;

model.Password = wfemp.Password;

}

else

{

ViewBag.Error = "Record not Found";

}

return View(model);

}

[HttpPost]

public ActionResult UpdateEmployee(AddemployeeModel model)

{

wf\_emp wfemp = db.wf\_emp.Where(i => i.Id == model.id).FirstOrDefault();

wfemp.Id = model.id;

wfemp.Employeeid = model.Eemployeeid;

wfemp.Name = model.Name;

wfemp.Dob = model.Dob;

wfemp.Gender = model.Gender;

wfemp.Address = model.Address;

wfemp.City = model.City;

wfemp.Email = model.Email;

wfemp.Mobile = model.Mobile;

wfemp.Experience = model.Experience;

wfemp.Password = model.Password;

db.SaveChanges();

return RedirectToAction("Employeelist", "Home");

}

[HttpGet]

public ActionResult RemoveEmployee(Int32 id)

{

wf\_emp wfemp = db.wf\_emp.Where(i => i.Id == id).FirstOrDefault();

db.wf\_emp.Remove(wfemp);

db.SaveChanges();

return RedirectToAction("Employeelist", "Home");

}

[HttpGet]

public ActionResult Emploggin()

{

ViewBag.Success = TempData["Success"];

return View();

}

[HttpPost]

public ActionResult Emploggin(EmplogginModel model)

{

if (ModelState.IsValid)

{

var account = db.wf\_emp.Where(i => i.Email == model.Email && i.Password == model.Password).FirstOrDefault();

if (account != null)

{

return RedirectToAction("Index", "Home");

}

else

{

TempData["Success"] = "Invalid User!";

return RedirectToAction("Emploggin", "Home");

}

}

else

{

TempData["Success"] = "All fields is required!";

return RedirectToAction("Emploggin", "Home");

}

}

[HttpGet]

public ActionResult Changepassword()

{

ViewBag.Success = TempData["Success"];

return View();

}

[HttpPost]

public ActionResult Changepassword(ChangepasswordModel model)

{

if (ModelState.IsValid)

{

var account = db.wf\_emp.Where(i => i.Email == model.Email && i.Password == model.Password).FirstOrDefault();

if (account != null)

{

wf\_emp wfemp = db.wf\_emp.Where(i => i.Email == model.Email).FirstOrDefault();

wfemp.Password = model.Newpassword;

db.SaveChanges();

TempData["Success"] = "You have change Password Successfully!!";

return RedirectToAction("Emploggin", "Home");

}

else

{

TempData["Success"] = "Invalid User!!";

return RedirectToAction("Changepassword", "Home");

}

}

else

{

TempData["Success"] = "All field is require!!";

return RedirectToAction("Changepassword", "Home");

}

}

[HttpGet]

public ActionResult Forgotpassword()

{

return View();

}

}

}

Add employee View

@model Workforce.Models.AddemployeeModel

@{

ViewBag.Title = "Addemployee";

Layout = "~/Views/\_Layout.cshtml";

}

<div class="container-fluid">

<div class="row">

<div class="col-auto">

<div class="page-title-box">

<h4 class="page-title float-left">Insert New Employee</h4>

<div class="clearfix"></div>

</div>

</div>

</div>

<!-- end row -->

<div class="row">

<div class="col-12">

<div class="card">

<div class="card-body">

<h4 class="m-t-0 header-title mb-4">Input Types</h4>

<div class="row">

<div class="col-12">

<div class="p-20">

@ViewBag.Success

@using (Html.BeginForm("Addemployee", "Home", FormMethod.Post, new { enctype = "multipart/form-data", @class = "form-horizontal", id = "Employee" }))

{

@Html.ValidationSummary(true,"", new { @class ="text-danger"})

<div class="form-group row">

<label class="col-2 col-form-label">Profile Picture</label>

<div class="fileupload fileupload-new" data-provides="fileupload">

<div class="col-lg-auto">

@Html.TextBoxFor(i => i.PhotoFile, new { type = "file" })

@Html.ValidationMessageFor(i => i.PhotoFile, "", new { @class = "text-danger" })

<a href="#" class="close fileupload-exists" data-dismiss="fileupload" style="float:inherit"><br />remove</a>

</div>

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">Resume</label>

<div class="fileupload fileupload-new" data-provides="fileupload">

<div class="col-lg-auto">

@Html.TextBoxFor(i => i.ResumeFile, new { type = "file" })

@Html.ValidationMessageFor(i => i.ResumeFile, "", new { @class = "text-danger" })

<a href="#" class="close fileupload-exists" data-dismiss="fileupload" style="float:inherit"><br />remove</a>

<br />

</div>

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">Employee ID</label>

<div class="col-lg-5">

@Html.TextBoxFor(i => i.Eemployeeid, new { @class = "form-control form-control-sm", placeholder = "Employee Id", id = "textname" })

@Html.ValidationMessageFor(i => i.Eemployeeid, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">Employee Name</label>

<div class="col-5">

@Html.TextBoxFor(i => i.Name, new { @class = "form-control form-control-sm", placeholder = "Name", id = "textname" })

@Html.ValidationMessageFor(i => i.Name, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">DOB</label>

<div class="col-5">

@Html.TextBoxFor(i => i.Dob, new { @class = "form-control form-control-sm", placeholder = "Dob", id = "textname", type = "date" })

@Html.ValidationMessageFor(i => i.Dob, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">Gender</label>

<div class="col-5">

<div class="mt-3">

<div class="custom-control custom-radio">

@Html.RadioButtonFor(i => i.Gender, "Male", new { @class = "custom-control-input", @id = "customRadio1" })

<label class="custom-control-label text-xs" for="customRadio1">Male</label>

</div>

<div class="custom-control custom-radio">

@Html.RadioButtonFor(i => i.Gender, "Female", new { @class = "custom-control-input", @id = "customRadio2" })

<label class="custom-control-label text-xs" for="customRadio2">Female</label>

</div>

</div>

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">Address</label>

<div class="col-5">

@Html.TextBoxFor(i => i.Address, new { @class = "form-control form-control-sm", placeholder = "Address", id = "textname" })

@Html.ValidationMessageFor(i => i.Address, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">City</label>

<div class="col-5">

@Html.DropDownListFor(i => i.City, ViewBag.Citylist as List<SelectListItem>, "Select City", new { @class = "custom-select mt-3" })

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">Email</label>

<div class="col-5">

@Html.TextBoxFor(i => i.Email, new { @class = "form-control form-control-sm", placeholder = "Email", id = "textname" })

@Html.ValidationMessageFor(i => i.Email, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">Mobile</label>

<div class="col-5">

@Html.TextBoxFor(i => i.Mobile, new { @class = "form-control form-control-sm", placeholder = "Mobile", id = "textname" })

@Html.ValidationMessageFor(i => i.Mobile, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">Experience</label>

<div class="col-5">

@Html.DropDownListFor(i => i.Experience, ViewBag.Experience as List<SelectListItem>, "Select Experience", new { @class = "custom-select mt-3" })

</div>

</div>

<div class="form-group row">

<label class="col-2 col-form-label">Password</label>

<div class="col-5">

@Html.PasswordFor(i => i.Password, new { @class = "form-control form-control-sm", placeholder = "Password", id = "textname" })

@Html.ValidationMessageFor(i => i.Password, "", new { @class = "text-danger" })

</div>

</div>

<input type="submit" value="Submit" class="btn btn-dark" id="btn">

}

</div>

</div>

</div>

<!-- end row -->

</div>

</div><!-- end card -->

</div><!-- end col -->

</div>

<!-- end row -->

</div>

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/jquery-migrate-1.2.1.min.js"></script>

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<link href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/2.0.1/css/toastr.css" rel="stylesheet" />

<script src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/2.0.1/js/toastr.js"></script>

<script src="~/Scripts/FileRemove.js"></script>

<link href="~/Content/Site.css" rel="stylesheet" />

<script type="text/javascript">

$(document).ready(function () {

toastr.options.timeOut = 3000

// show when page load

toastr.info('Page Loaded!');

$('#btn').click(function () {

// show when the button is clicked

toastr.success('Employee Added Successfully');

});

});

</script>

Addemployee Model

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

using System.Web.Mvc;

using System.IO;

namespace Workforce.Models

{

public class AddemployeeModel

{

public int id { get; set; }

[Required(ErrorMessage = "Please select file.")]

[RegularExpression(@"([a-zA-Z0-9\s\_\\.\-:])+(.png|.jpg|.gif)$", ErrorMessage = "Only Image files allowed.")]

public HttpPostedFileBase PhotoFile { get; set; }

public string Photo { get; set; }

[Required(ErrorMessage = "Please select file.")]

[RegularExpression(@"([a-zA-Z0-9\s\_\\.\-:])+(.doc|.docx|.pdf)$", ErrorMessage = "Only doc files allowed.")]

public HttpPostedFileBase ResumeFile { get; set; }

public string Resume { get; set; }

[Required]

public int? Eemployeeid { get; set; }

[Required]

[RegularExpression(@"^(([A-za-z]+[\s]{1}[A-za-z]+)|([A-Za-z]+))$", ErrorMessage = "Only Alphabets are allowed")]

public string Name { get; set; }

[Required]

[DataType(DataType.DateTime)]

[DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:d}")]

public DateTime? Dob { get; set; }

[Required]

public string Gender { get; set; }

[Required]

public string Address { get; set; }

[Required]

public string City { get; set; }

[Required]

[EmailAddress]

[Remote("IsEmailAvailable", "Home",ErrorMessage =("This User allready Added"))]

public string Email { get; set; }

[Required]

[StringLength(10,ErrorMessage = "maximum 10 digit only",MinimumLength =10)]

public string Mobile { get; set; }

[Required]

public string Experience { get; set; }

[Required]

public string Password { get; set; }

}

}

Online

@model Onlinestore.Models.CategoryModel

@{

ViewBag.Title = "CreateCategory";

}

<h2>CreateCategory</h2>

<div class="row">

<div class="col-lg-auto">

@ViewBag.Success

@using (Html.BeginForm("CreateCategory", "Admin", FormMethod.Post, new {enctype = "multipart/form-data" }))

{

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

<label><b>Category Name:</b></label>

@Html.TextBoxFor(i => i.CategoryName, new { @class = "form-control", placeholder = "Category Name", id = "textname" })

@Html.ValidationMessageFor(i => i.CategoryName, "", new { @class = "text-danger" })

</div>

<div class="form-group">

<label><b>Category Image:</b></label>

@Html.TextBoxFor(i => i.CategoryImage, new { type = "file" })

@Html.ValidationMessageFor(i => i.CategoryImage, "", new { @class = "text-danger" })

</div><br />

<input type="submit" value="Create" />

}

</div>

</div>