

Page: Registration form and employee records:

1.Sql query:

```
create table department(  
id int identity,  
dept_id int primary key,  
dept_name nvarchar(20)  
)  
insert into department values(101,'Advisory'),(102,'Audit'),(103,'Tax'),(104,'Consulting')  
select * from department
```

```
create table basic(  
  
empid int primary key,  
empname nvarchar(20),  
emptype nvarchar(20),  
deptid int foreign key references department(dept_id),  
dob date,  
email nvarchar(20),  
mobile nvarchar(20),  
address nvarchar(20),  
dateofjoin date,  
gender varchar(20),  
emp_photo_path varchar(max)  
)
```

```
create table education(  
id int primary key identity,  
empid int foreign key references basic(empid),  
Tenth_percentage nvarchar(20),  
Tenth_yop int,  
Twelfth_percentage nvarchar(20),  
Twelfth_yop int,  
UG_percentage nvarchar(20),  
UG_yop int,  
PG_percentage nvarchar(20),  
PG_yop nvarchar(20)  
)
```

```
create table work  
(id int primary key identity,  
empid int foreign key references basic(empid),
```

```
previous_emp_comp_name nvarchar(10),  
year_exp nvarchar(20),  
work_domain nvarchar(20),  
)
```

```
create table skills  
(id int primary key identity,  
empid int foreign key references basic(empid),  
primary_skills nvarchar(20),  
other_skills nvarchar(20)  
)
```

```
select * from department  
select * from basic  
select * from education  
select * from work  
select * from skills
```

2.VS CODE:

HOME CONTROLLER: (Add a empty folder named “Image” in solution explorer)

Note:Add 2 lines in “basic.cs” model file(mentioned below)

:No need to add view for delete as we are not using it,just have the code alone in controller

Basic.cs (Add highlighted lines at that place)

```
public string gender { get; set; }  
[DisplayName("Upload Photo")]  
public string emp_photo_path { get; set; }  
public HttpPostedFileBase ImageFile { get; set; }
```

CONTROLLER CODE:

```
public class HomeController : Controller
```

```
{  
    private project1Entities p = new project1Entities();
```

//BASIC DETAILS TABLE

//Basic details create template

```
public ActionResult basic()  
{  
    return View();  
}  
[HttpPost]  
public ActionResult basic(basic obj)  
{  
    string filename = Path.GetFileNameWithoutExtension(obj.ImageFile.FileName);  
    string extension = Path.GetExtension(obj.ImageFile.FileName);  
    filename = filename + DateTime.Now.ToString("yymmssfff") + extension;  
    obj.emp_photo_path = "~/Image/" + filename;  
    filename = Path.Combine(Server.MapPath("~/Image/"), filename);  
    obj.ImageFile.SaveAs(filename);  
  
    using (project1Entities p = new project1Entities())  
    {  
        if (ModelState.IsValid)  
        {  
            p.basics.Add(obj);  
            p.SaveChanges();  
            return RedirectToAction("education");  
        }  
    }  
    ModelState.Clear();  
  
    return View(obj);  
}
```

//Basic details in list template

```
public ActionResult basiclist()  
{  
    project1Entities p = new project1Entities();  
    List<basic> b = p.basics.ToList();
```

```

    return View(b);
}

```

//basic details in details template

```

public ActionResult Details(int id)
{
    project1Entities p = new project1Entities();
    basic b = p.basics.FirstOrDefault(a => a.empid == id);
    return View(b);
}

```

//Basic details in edit template

```

public ActionResult Edit(int? id)
{
    project1Entities p = new project1Entities();
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    basic basic = p.basics.Find(id);
    if (basic == null)
    {
        return HttpNotFound();
    }
    ViewBag.deptid = new SelectList(p.departments, "dept_id", "dept_name", basic.deptid);
    return View(basic);
}

```

// POST: basics/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to,
for

// more details see <https://go.microsoft.com/fwlink/?LinkId=317598>.

[HttpPost]

[ValidateAntiForgeryToken]

```

public ActionResult Edit([Bind(Include =
"empid,empname,emptype,deptid,dob,email,mobile,address,dateofjoin,gender,emp_photo_path
")] basic basic)

```

```

{
    project1Entities p = new project1Entities();
    if (ModelState.IsValid)

```

```

    {
        p.Entry(basic).State = EntityState.Modified;
        p.SaveChanges();
        return RedirectToAction("basiclist");
    }
    ViewBag.deptid = new SelectList(p.departments, "dept_id", "dept_name", basic.deptid);
    return View(basic);
}

```

//Basic details in delete template

```

public ActionResult Delete(int? id)
{
    project1Entities p = new project1Entities();
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    basic basic = p.basics.Find(id);
    if (basic == null)
    {
        return HttpNotFound();
    }
    return View(basic);
}

```

```

// POST: basics/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    project1Entities p = new project1Entities();
    basic basic = p.basics.Find(id);
    p.basics.Remove(basic);
    p.SaveChanges();
    return RedirectToAction("basiclist");
}

```

//EDUCATION DETAILS TABLE

//education details in create template

```
public ActionResult education()
{
    return View();
}
[HttpPost]
public ActionResult education(education obj)
{
    project1Entities p = new project1Entities();
    if (ModelState.IsValid)
    {
        p.educations.Add(obj);
        p.SaveChanges();
        return RedirectToAction("work");
    }
    return View(obj);
}
```

//education details in details template

```
public ActionResult edudetails(int id)
{
    project1Entities p = new project1Entities();
    education e = p.educations.FirstOrDefault(a => a.empid == id);
    return View(e);
}
```

//education details in edit template

```
public ActionResult Edit1(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    education education = p.educations.Find(id);
    if (education == null)
    {
        return HttpNotFound();
    }
}
```

```

    }
    ViewBag.empid = new SelectList(p.basics, "empid", "empname", education.empid);
    return View(education);
}

// POST: educations/Edit/5
// To protect from overposting attacks, enable the specific properties you want to bind to,
for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit1([Bind(Include =
"id,empid,Tenth_percentage,Tenth_yop,Twelfth_percentage,Twelfth_yop,UG_percentage,UG_y
op,PG_percentage,PG_yop")] education education)
{
    if (ModelState.IsValid)
    {
        p.Entry(education).State = EntityState.Modified;
        p.SaveChanges();
        return RedirectToAction("basiclist");
    }
    ViewBag.empid = new SelectList(p.basics, "empid", "empname", education.empid);
    return View(education);
}

```

//education details in delete template

```

public ActionResult Delete1(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    education education = p.educations.Find(id);
    if (education == null)
    {
        return HttpNotFound();
    }
    return View(education);
}

// POST: educations/Delete/5
[HttpPost, ActionName("Delete1")]
[ValidateAntiForgeryToken]

```

```

public ActionResult DeleteConfirmed1(int id)
{
    education education = p.educations.Find(id);
    p.educations.Remove(education);
    p.SaveChanges();
    return RedirectToAction("basiclist");
}

```

//WORK DETAILS TABLE

//Work details in create template

```

public ActionResult work()
{
    return View();
}
[HttpPost]
public ActionResult work(work obj)
{
    project1Entities p = new project1Entities();
    if (ModelState.IsValid)
    {
        p.works.Add(obj);
        p.SaveChanges();
        return RedirectToAction("skill");
    }
    return View(obj);
}

```

//workdetails in details template

```

public ActionResult workdetails(int id)
{
    project1Entities p = new project1Entities();
    work w = p.works.FirstOrDefault(a => a.empid == id);
    return View(w);
}

```


//Work details in edit template

```
public ActionResult Edit2(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    work work = p.works.Find(id);
    if (work == null)
    {
        return HttpNotFound();
    }
    ViewBag.empid = new SelectList(p.basics, "empid", "empname", work.empid);
    return View(work);
}
```

// POST: works/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to,

for

// more details see <https://go.microsoft.com/fwlink/?LinkId=317598>.

[HttpPost]

[ValidateAntiForgeryToken]

```
public ActionResult Edit2([Bind(Include =
"id,empid,previous_emp_comp_name,year_exp,work_domain")] work work)
```

```
{
    if (ModelState.IsValid)
    {
        p.Entry(work).State = EntityState.Modified;
        p.SaveChanges();
        return RedirectToAction("basiclist");
    }
    ViewBag.empid = new SelectList(p.basics, "empid", "empname", work.empid);
    return View(work);
}
```

//work details in delete template

```
public ActionResult Delete2(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
}
```

```

    }
    work work = p.works.Find(id);
    if (work == null)
    {
        return HttpNotFound();
    }
    return View(work);
}

// POST: works/Delete/5
[HttpPost, ActionName("Delete2")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed2(int id)
{
    work work = p.works.Find(id);
    p.works.Remove(work);
    p.SaveChanges();
    return RedirectToAction("basiclist");
}

```

//SKILL DETAILS TABLE

```

//skill details in create template
public ActionResult skill()
{
    return View();
}

[HttpPost]
public ActionResult skill(skill obj)
{
    project1Entities p = new project1Entities();
    if (ModelState.IsValid)
    {
        p.skills.Add(obj);
        p.SaveChanges();
        return RedirectToAction("basic");
    }
    return View(obj);
}

```

//skill details in details template

```
public ActionResult skilldetails(int id)
{
    project1Entities p = new project1Entities();
    skill s = p.skills.FirstOrDefault(a => a.empid == id);
    return View(s);
}
```

//skill details in edit template

```
public ActionResult Edit3(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    skill skill = p.skills.Find(id);
    if (skill == null)
    {
        return HttpNotFound();
    }
    ViewBag.empid = new SelectList(p.basics, "empid", "empname", skill.empid);
    return View(skill);
}
```

// POST: skills/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to,
for

// more details see <https://go.microsoft.com/fwlink/?LinkId=317598>.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit3([Bind(Include = "id,empid,primary_skills,other_skills")] skill skill)

```
{
    if (ModelState.IsValid)
    {
        p.Entry(skill).State = EntityState.Modified;
        p.SaveChanges();
        return RedirectToAction("basiclist");
    }
}
```

```

        ViewBag.empid = new SelectList(p.basics, "empid", "empname", skill.empid);
        return View(skill);
    }

```

//skill details in delete template

```

public ActionResult Delete3(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    skill skill = p.skills.Find(id);
    if (skill == null)
    {
        return HttpNotFound();
    }
    return View(skill);
}

```

```

// POST: skills/Delete/5
[HttpPost, ActionName("Delete3")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed3(int id)
{
    skill skill = p.skills.Find(id);
    p.skills.Remove(skill);
    p.SaveChanges();
    return RedirectToAction("basiclist");
}

```

Basic.cshtml-----view

```
@model project1.Models.basic
```

```
@{  
    ViewBag.Title = "basic";  
}
```

```
<h2>basic</h2>
```

```
@using (Html.BeginForm("basic", "Home", FormMethod.Post, new { enctype =  
"multipart/form-data" }))  
{  
    @Html.AntiForgeryToken()  
  
    <div class="form-horizontal">  
        <h4>basic</h4>  
        <hr />  
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })  
        <div class="form-group">  
            @Html.LabelFor(model => model.empid, htmlAttributes: new { @class = "control-label  
col-md-2" })  
            <div class="col-md-10">  
                @Html.EditorFor(model => model.empid, new { htmlAttributes = new { @class =  
"form-control" } })  
                @Html.ValidationMessageFor(model => model.empid, "", new { @class =  
"text-danger" })  
            </div>  
        </div>  
  
        <div class="form-group">  
            @Html.LabelFor(model => model.empname, htmlAttributes: new { @class =  
"control-label col-md-2" })  
            <div class="col-md-10">  
                @Html.EditorFor(model => model.empname, new { htmlAttributes = new { @class =  
"form-control" } })  
                @Html.ValidationMessageFor(model => model.empname, "", new { @class =  
"text-danger" })  
            </div>  
        </div>  
  
        <div class="form-group">
```

```

        @Html.LabelFor(model => model.emptytype, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.emptytype, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.emptytype, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.deptid, "deptid", htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.deptid, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.deptid, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.dob, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.dob, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.dob, "", new { @class = "text-danger"
})
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.email, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.email, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.email, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">

```

```

        @Html.LabelFor(model => model.mobile, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.mobile, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.mobile, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.address, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.address, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.address, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.dateofjoin, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.dateofjoin, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.dateofjoin, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.gender, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.gender, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.gender, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">

```

```
        @Html.LabelFor(model => model.emp_photo_path, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            <input type="file" name="ImageFile" required />
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Create" class="btn btn-default" />
        </div>
    </div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```


Basiclist.cshtml

```
@model IEnumerable<project1.Models.basic>
```

```
@{  
    ViewBag.Title = "basiclist";  
}
```

```
<h2>basiclist</h2>
```

```
@*<p>  
    @Html.ActionLink("Create New", "Create")  
</p>*@  
<table class="table">  
    <tr>  
        <th>  
            @Html.DisplayNameFor(model => model.empid)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.empname)  
        </th>  
        @*<th>  
            @Html.DisplayNameFor(model => model.emptype)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.dob)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.email)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.mobile)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.address)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.dateofjoin)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.gender)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.emp_photo_path)  
        </th>
```

```

        <th>
            @Html.DisplayNameFor(model => model.department.dept_name)
        </th>*@
    <th></th>
</tr>

```

```

@foreach (var item in Model) {

```

```

    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.empid)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.empname)
        </td>
        @*<td>
            @Html.DisplayFor(modelItem => item.emptype)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.dob)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.email)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.mobile)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.address)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.dateofjoin)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.gender)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.emp_photo_path)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.department.dept_name)
        </td>*@
    <td>
        @*@Html.ActionLink("Edit", "Edit", new { id = item.empid }) |*@
        @Html.ActionLink("Basic Details", "Details", new { id = item.empid })
    </td>

```

```
        @*@Html.ActionLink("Delete", "Delete", new { id = item.empid })*@
    </td>

    <td>
        @Html.ActionLink("Education Details", "edudetails", new { id = item.empid })
    </td>
    <td>
        @Html.ActionLink("Work Experience Details", "workdetails", new { id = item.empid })
    </td>
    <td>
        @Html.ActionLink("Skill Details", "skilldetails", new { id = item.empid })
    </td>
</tr>
}

</table>
```

Details.cshtml

@model project1.Models.basic

```
@{  
    ViewBag.Title = "Details";  
}
```

<h2>Details</h2>

<div>

<h4>basic</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.empid)

</dt>

<dd>

@Html.DisplayFor(model => model.empid)

</dd>

<dt>

@Html.DisplayNameFor(model => model.empname)

</dt>

<dd>

@Html.DisplayFor(model => model.empname)

</dd>

<dt>

@Html.DisplayNameFor(model => model.emptype)

</dt>

<dd>

@Html.DisplayFor(model => model.emptype)

</dd>

<dt>

@Html.DisplayNameFor(model => model.dob)

</dt>

<dd>

@Html.DisplayFor(model => model.dob)

</dd>

```
<dt>
    @Html.DisplayNameFor(model => model.email)
</dt>

<dd>
    @Html.DisplayFor(model => model.email)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.mobile)
</dt>

<dd>
    @Html.DisplayFor(model => model.mobile)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.address)
</dt>

<dd>
    @Html.DisplayFor(model => model.address)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.dateofjoin)
</dt>

<dd>
    @Html.DisplayFor(model => model.dateofjoin)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.gender)
</dt>

<dd>
    @Html.DisplayFor(model => model.gender)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.emp_photo_path)
</dt>
```

```

<dd>
    
</dd>

<dt>
    @Html.DisplayNameFor(model => model.department.dept_name)
</dt>

<dd>
    @Html.DisplayFor(model => model.department.dept_name)
</dd>

</dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.empid }) |
    @Html.ActionLink("Back to List", "basiclist")
</p>

```

Edit.cshtml

@model project1.Models.basic

```

@{
    ViewBag.Title = "Edit";
}

```

<h2>Edit</h2>

```

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>basic</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.empid)

        <div class="form-group">

```

```

        @Html.LabelFor(model => model.empname, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.empname, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.empname, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.emptype, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.emptype, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.emptype, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.deptid, "deptid", htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.deptid, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.deptid, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.dob, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.dob, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.dob, "", new { @class = "text-danger"
})
        </div>
    </div>

    <div class="form-group">

```

```

        @Html.LabelFor(model => model.email, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.email, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.email, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.mobile, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.mobile, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.mobile, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.address, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.address, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.address, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.dateofjoin, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.dateofjoin, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.dateofjoin, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">

```



```

        @Html.LabelFor(model => model.gender, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.gender, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.gender, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.emp_photo_path, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.emp_photo_path, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.emp_photo_path, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Save" class="btn btn-default" />
        </div>
    </div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "basiclist")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

Education.cshtml

@model project1.Models.education

```
@{
    ViewBag.Title = "education";
}
```

<h2>education</h2>

@using (Html.BeginForm())

```
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>education</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.empid, "empid", htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.empid, new { htmlAttributes = new { @class =
"form-control" } })
                @Html.ValidationMessageFor(model => model.empid, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Tenth_percentage, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Tenth_percentage, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Tenth_percentage, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
```

```

        @Html.LabelFor(model => model.Tenth_yop, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Tenth_yop, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.Tenth_yop, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Twelfth_percentage, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Twelfth_percentage, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.Twelfth_percentage, "", new { @class
= "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Twelfth_yop, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Twelfth_yop, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.Twelfth_yop, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.UG_percentage, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.UG_percentage, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.UG_percentage, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">

```

```

        @Html.LabelFor(model => model.UG_yop, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.UG_yop, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.UG_yop, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.PG_percentage, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.PG_percentage, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.PG_percentage, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.PG_yop, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.PG_yop, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.PG_yop, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Create" class="btn btn-default" />
        </div>
    </div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

```

```
@section Scripts {  
    @Scripts.Render("~/bundles/jqueryval")  
}
```

Edudetails.cshtml

```
@model project1.Models.education
```

```
@{  
    ViewBag.Title = "edudetails";  
}
```

```
<h2>edudetails</h2>
```

```
<div>
```

```
    <h4>education</h4>
```

```
    <hr />
```

```
    <dl class="dl-horizontal">
```

```
        <dt>
```

```
        @Html.DisplayNameFor(model => model.empid)
```

```
        </dt>
```

```
        <dd>
```

```
        @Html.DisplayFor(model => model.empid)
```

```
        </dd>
```

```
        <dt>
```

```
        @Html.DisplayNameFor(model => model.Tenth_percentage)
```

```
        </dt>
```

```
        <dd>
```

```
        @Html.DisplayFor(model => model.Tenth_percentage)
```

```
        </dd>
```

```
        <dt>
```

```
        @Html.DisplayNameFor(model => model.Tenth_yop)
```

```
        </dt>
```

```
        <dd>
```

```
    @Html.DisplayFor(model => model.Tenth_yop)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.Twelfth_percentage)
</dt>

<dd>
    @Html.DisplayFor(model => model.Twelfth_percentage)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.Twelfth_yop)
</dt>

<dd>
    @Html.DisplayFor(model => model.Twelfth_yop)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.UG_percentage)
</dt>

<dd>
    @Html.DisplayFor(model => model.UG_percentage)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.UG_yop)
</dt>

<dd>
    @Html.DisplayFor(model => model.UG_yop)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.PG_percentage)
</dt>

<dd>
    @Html.DisplayFor(model => model.PG_percentage)
</dd>

<dt>
```

```

        @Html.DisplayNameFor(model => model.PG_yop)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.PG_yop)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.basic.empname)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.basic.empname)
    </dd>

</dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit1", new { id = Model.id }) |
    @* @Html.ActionLink("Delete", "Delete1", new {id=Model.id}) |
    @Html.ActionLink("Create","education1") |* @
    @Html.ActionLink("Back to List", "basiclist")
</p>

```

Edit1.cshtml

```
@model project1.Models.education
```

```
@{
    ViewBag.Title = "Edit1";
}
```

```
<h2>Edit1</h2>
```

```
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
```

```

<h4>education</h4>
<hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model => model.id)

<div class="form-group">
    @Html.LabelFor(model => model.empid, "empid", htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.empid, new { htmlAttributes = new { @class =
"form-control" } })
        @Html.ValidationMessageFor(model => model.empid, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Tenth_percentage, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Tenth_percentage, new { htmlAttributes = new {
@class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Tenth_percentage, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Tenth_yop, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Tenth_yop, new { htmlAttributes = new { @class =
"form-control" } })
        @Html.ValidationMessageFor(model => model.Tenth_yop, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Twelfth_percentage, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Twelfth_percentage, new { htmlAttributes = new {
@class = "form-control" } })

```



```

        @Html.ValidationMessageFor(model => model.Twelfth_percentage, "", new { @class
= "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Twelfth_yop, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Twelfth_yop, new { htmlAttributes = new { @class =
"form-control" } })
        @Html.ValidationMessageFor(model => model.Twelfth_yop, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.UG_percentage, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.UG_percentage, new { htmlAttributes = new {
@class = "form-control" } })
        @Html.ValidationMessageFor(model => model.UG_percentage, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.UG_yop, htmlAttributes: new { @class = "control-label
col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.UG_yop, new { htmlAttributes = new { @class =
"form-control" } })
        @Html.ValidationMessageFor(model => model.UG_yop, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.PG_percentage, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.PG_percentage, new { htmlAttributes = new {
@class = "form-control" } })

```

```

        @Html.ValidationMessageFor(model => model.PG_percentage, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.PG_yop, htmlAttributes: new { @class = "control-label
col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.PG_yop, new { htmlAttributes = new { @class =
"form-control" } })
        @Html.ValidationMessageFor(model => model.PG_yop, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Save" class="btn btn-default" />
    </div>
</div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "basiclist")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

Work.cshtml

```
@model project1.Models.work
```

```
@{  
    ViewBag.Title = "work";  
}
```

```
<h2>work</h2>
```

```
@using (Html.BeginForm())
```

```
{  
    @Html.AntiForgeryToken()
```

```
    <div class="form-horizontal">
```

```
        <h4>work</h4>
```

```
        <hr />
```

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

```
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.empid, "empid", htmlAttributes: new { @class =  
"control-label col-md-2" })
```

```
            <div class="col-md-10">
```

```
                @Html.EditorFor(model => model.empid, new { htmlAttributes = new { @class =  
"form-control" } })
```

```
                @Html.ValidationMessageFor(model => model.empid, "", new { @class =  
"text-danger" })
```

```
            </div>
```

```
        </div>
```

```
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.previous_emp_comp_name, htmlAttributes: new {  
@class = "control-label col-md-2" })
```

```
            <div class="col-md-10">
```

```
                @Html.EditorFor(model => model.previous_emp_comp_name, new { htmlAttributes =  
new { @class = "form-control" } })
```

```
                @Html.ValidationMessageFor(model => model.previous_emp_comp_name, "", new {  
@class = "text-danger" })
```

```
            </div>
```

```
        </div>
```

```
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.year_exp, htmlAttributes: new { @class =  
"control-label col-md-2" })
```

```

        <div class="col-md-10">
            @Html.EditorFor(model => model.year_exp, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.year_exp, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.work_domain, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.work_domain, new { htmlAttributes = new { @class
= "form-control" } })
            @Html.ValidationMessageFor(model => model.work_domain, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Create" class="btn btn-default" />
        </div>
    </div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

Workdetails.cshtml

@model project1.Models.work

```
@{  
    ViewBag.Title = "workdetails";  
}
```

<h2>workdetails</h2>

<div>

<h4>work</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.previous_emp_comp_name)

</dt>

<dd>

@Html.DisplayFor(model => model.previous_emp_comp_name)

</dd>

<dt>

@Html.DisplayNameFor(model => model.year_exp)

</dt>

<dd>

@Html.DisplayFor(model => model.year_exp)

</dd>

<dt>

@Html.DisplayNameFor(model => model.work_domain)

</dt>

<dd>

@Html.DisplayFor(model => model.work_domain)

</dd>

<dt>

@Html.DisplayNameFor(model => model.basic.empname)

</dt>

<dd>

@Html.DisplayFor(model => model.basic.empname)

```

        </dd>

    </dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit2", new { id = Model.id }) |
    @Html.ActionLink("Back to List", "basiclist")
</p>

```

Edit2.cshtml

@model project1.Models.work

```

@{
    ViewBag.Title = "Edit2";
}

```

<h2>Edit2</h2>

```

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>work</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.id)

        <div class="form-group">
            @Html.LabelFor(model => model.empid, "empid", htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.empid, new { htmlAttributes = new { @class =
"form-control" } })
                @Html.ValidationMessageFor(model => model.empid, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">

```

```

        @Html.LabelFor(model => model.previous_emp_comp_name, htmlAttributes: new {
@class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.previous_emp_comp_name, new { htmlAttributes =
new { @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.previous_emp_comp_name, "", new {
@class = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.year_exp, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.year_exp, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.year_exp, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.work_domain, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.work_domain, new { htmlAttributes = new { @class
= "form-control" } })
            @Html.ValidationMessageFor(model => model.work_domain, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Save" class="btn btn-default" />
        </div>
    </div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "basiclist")
</div>

```

```
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Skill.cshtml

```
@model project1.Models.skill
```

```
@{
    ViewBag.Title = "skill";
}
```

```
<h2>skill</h2>
```

```
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>skill</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.empid, "empid", htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.empid, new { htmlAttributes = new { @class =
"form-control" } })
                @Html.ValidationMessageFor(model => model.empid, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.primary_skills, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
```



```

        @Html.EditorFor(model => model.primary_skills, new { htmlAttributes = new { @class
= "form-control" } })
        @Html.ValidationMessageFor(model => model.primary_skills, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.other_skills, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.other_skills, new { htmlAttributes = new { @class =
"form-control" } })
        @Html.ValidationMessageFor(model => model.other_skills, "", new { @class =
"text-danger" })
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Create" class="btn btn-default" />
    </div>
</div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

Skilldetails.cshtml

@model project1.Models.skill

```
@{
    ViewBag.Title = "skilldetails";
}
```

<h2>skilldetails</h2>

<div>

<h4>skill</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.primary_skills)

</dt>

<dd>

@Html.DisplayFor(model => model.primary_skills)

</dd>

<dt>

@Html.DisplayNameFor(model => model.other_skills)

</dt>

<dd>

@Html.DisplayFor(model => model.other_skills)

</dd>

<dt>

@Html.DisplayNameFor(model => model.basic.empname)

</dt>

<dd>

@Html.DisplayFor(model => model.basic.empname)

</dd>

</dl>

</div>

<p>

@Html.ActionLink("Edit", "Edit3", new { id = Model.id }) |

@Html.ActionLink("Back to List", "basiclist")

</p>

Edit3.cshtml

@model project1.Models.skill

```
@{
    ViewBag.Title = "Edit3";
}
```

<h2>Edit3</h2>

```
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>skill</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.id)

        <div class="form-group">
            @Html.LabelFor(model => model.empid, "empid", htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.empid, new { htmlAttributes = new { @class =
"form-control" } })
                @Html.ValidationMessageFor(model => model.empid, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.primary_skills, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.primary_skills, new { htmlAttributes = new { @class
= "form-control" } })
                @Html.ValidationMessageFor(model => model.primary_skills, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
```

```

        @Html.LabelFor(model => model.other_skills, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.other_skills, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.other_skills, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Save" class="btn btn-default" />
        </div>
    </div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "basiclist")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

PAGE:Department list

departmentController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Emp.Models;

namespace Emp.Controllers
{
    public class departmentController : Controller
    {
        // GET: Department
        public ActionResult Index()
        {
            EmployeeManagementSystemEntities db = new
EmployeeManagementSystemEntities();
            List<department> Id = db.departments.ToList();

            return View(Id);
        }
    }
}
```

dept_employeeController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Emp.Models;
```

```

namespace Emp.Controllers
{
    public class dept_employeeController : Controller
    {
        // GET: dept_employee
        public ActionResult Index(int dept_id)
        {
            EmployeeManagementSystemEntities db = new
EmployeeManagementSystemEntities();
            List<basic> st = db.basics.Where(s => s.deptid == dept_id).ToList();
            return View(st);
        }
        public ActionResult Details(int id)
        {
            EmployeeManagementSystemEntities db = new
EmployeeManagementSystemEntities();
            basic st = db.basics.FirstOrDefault(x => x.empid == id);
            return View(st);
        }
    }
}

```

index.cshtml of departmentController

```

@model IEnumerable<Emp.Models.department>
@using Emp.Models;

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

```

```
<ul>
```

```
    @foreach (department Dept in @Model)
```

```
    {
```

```
        <li>
```

```
            @Html.ActionLink(Dept.dept_name, "Index", "dept_employee", new { dept_id =  
Dept.dept_id }, null)
```

```
        </li>
```

```
    }
```

```
</ul>
```

Index.cshtml of dept_employeeController

```
@model IEnumerable<Emp.Models.basic>
```

```
@using Emp.Models;
```

```
@{
```

```
    ViewBag.Title = "Index";
```

```
}
```

```
<div>
```

```
    <h2>StudentList</h2>
```

```
    <ul>
```

```
        @foreach (basic st in @Model)
```

```
        {
```

```
            <li>
```

```
                @Html.ActionLink(st.empname, "Details", new { id = st.empid })
```

```
            </li>
```

```
        }
```

```
    </ul>
```

```
    @Html.ActionLink("back to branch list", "Index", "department")
```

```
</div>
```

Details.cshtml of dept_employeeController

```
@model Emp.Models.basic
```

```
@{  
    ViewBag.Title = "Details";  
}
```

```
<h2>Details</h2>
```

```
<div>
```

```
    <h4>basic</h4>
```

```
    <hr />
```

```
    <dl class="dl-horizontal">
```

```
        <dt>
```

```
            @Html.DisplayNameFor(model => model.empname)
```

```
        </dt>
```

```
        <dd>
```

```
            @Html.DisplayFor(model => model.empname)
```

```
        </dd>
```

```
        <dt>
```

```
            @Html.DisplayNameFor(model => model.emptype)
```

```
        </dt>
```

```
        <dd>
```

```
            @Html.DisplayFor(model => model.emptype)
```

```
        </dd>
```

```
        <dt>
```

```
            @Html.DisplayNameFor(model => model.dob)
```

```
        </dt>
```

```
        <dd>
```

```
            @Html.DisplayFor(model => model.dob)
```

```
        </dd>
```

```
        <dt>
```

```
            @Html.DisplayNameFor(model => model.email)
```



```
</dt>

<dd>
    @Html.DisplayFor(model => model.email)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.mobile)
</dt>

<dd>
    @Html.DisplayFor(model => model.mobile)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.address)
</dt>

<dd>
    @Html.DisplayFor(model => model.address)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.dateofjoin)
</dt>

<dd>
    @Html.DisplayFor(model => model.dateofjoin)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.gender)
</dt>

<dd>
    @Html.DisplayFor(model => model.gender)
</dd>

<dt>
    
</dt>

<dd>
    @Html.DisplayFor(model => model.emp_photo_path)
```

```
</dd>

<dt>
    @Html.DisplayNameFor(model => model.department.dept_name)
</dt>

<dd>
    @Html.DisplayFor(model => model.department.dept_name)
</dd>

</dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.empid }) |
    @Html.ActionLink("Back to List", "Index")
</p>
```

PAGE:LOGIN SIGNUP

1.SQL query

```
create table login(id int not null primary key identity, emailid nvarchar(40),password  
nvarchar(20))
```

```
create table signup(id int not null primary key identity, name nvarchar(40), empid nvarchar(40),  
mobileno nvarchar(15),dob date,emailid nvarchar(40),password nvarchar(20),  
confirmpassword nvarchar(20),securityquestion nvarchar(100),securityanswer nvarchar(100))
```

```
create table forgot(id int not null primary key identity, emailid nvarchar(40),  
securityquestion nvarchar(100),securityanswer nvarchar(100),newpassword nvarchar(20),  
confirmpassword nvarchar(20))
```

```
select * from signup  
select * from login  
select * from forgot
```

2.vs code

HomeController.cs

```
public class HomeController : Controller  
{  
    private project1Entities m = new project1Entities();  
  
    //index is the login page  
    public ActionResult Index()  
    {  
        return View();  
    }  
    [HttpPost]  
    public ActionResult Index(login log)  
    {  
        if (this.IsCaptchaValid("Captcha is not valid"))  
        {  
            bool isvaliduser = m.signups.Any(x => x.emailid == log.emailid && x.password ==  
log.password);  
            if (isvaliduser)  
            {  
                FormsAuthentication.SetAuthCookie(log.emailid, false);
```

```

        m.logins.Add(log);
        m.SaveChanges();
        int id = log.id;
        return RedirectToAction("basic","Register"); //Sasmithaa will put the employee path

    }
    else
    {
        ModelState.AddModelError("", "Invalid emailid or Password");
    }
}
ViewBag.ErrMessage = "Error: Invalid Captcha";

return View();

}

public ActionResult signup()
{
    return View();
}
[HttpPost]
public ActionResult signup(signup sign)
{
    if (this.IsCaptchaValid("Captcha is not valid"))
    {
        bool isvalid = m.signups.Any(x => x.emailid != sign.emailid);
        if (isvalid)
        {
            FormsAuthentication.SetAuthCookie(sign.emailid, false);
            m.signups.Add(sign);
            m.SaveChanges();
            int id = sign.id;

            return RedirectToAction("Index");
        }
    }
    else
    {
        ModelState.AddModelError("", "Email already Exists");
    }
}

```

```

    }
}
ViewBag.ErrorMessage = "Error: Invalid Captcha";
ViewBag.Message = "Thanks for accepting Terms and Conditions";
return View();
}

public ActionResult forgot()
{
    return View();
}
[HttpPost]
public ActionResult forgot(forgot forgot)
{
    bool isvalid = m.signups.Any(x => x.emailid == forgot.emailid);
    bool valid = m.signups.Any(x => x.securityquestion == forgot.securityquestion &&
x.securityanswer.ToLower() == forgot.securityanswer.ToLower());
    if (valid)
    {
        if (isvalid)
        {
            var pass = m.signups.Where(x => x.emailid.Equals(forgot.emailid));
            foreach (signup si in pass)
            {
                si.password = forgot.newpassword;
                si.confirmpassword = forgot.confirmpassword;
            }
            m.forgots.Add(forgot);
            m.SaveChanges();
            return RedirectToAction("Index");
        }
        ModelState.AddModelError("", "Email doesn't exists");
    }
    ModelState.AddModelError("", "Answered Wrong for the Security Question");

    return View();
}
}

```

Index.cshtml

```
@model project1.Models.login
@using CaptchaMvc.HtmlHelpers;
@using project1.Models;
```

```
@{
    ViewBag.Title = "Index";
}
<h2>Index</h2>
```

```
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>login</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.emailid, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.emailid, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.emailid, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.password, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.password, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.password, "", new { @class = "text-danger" })
            </div>
        </div>
    </div>
}
```

```

</div>

<div>
    @Html.Captcha(5)

    <p class="Error">@ViewBag.ErrMessage</p>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Login" class="btn btn-default" />
    </div>
</div>
</div>
}

<div>
    @Html.ActionLink("Forgot Password", "forgot")
</div>
<div>
    @Html.ActionLink("New User? Sign Up", "signup")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

Signup.cshtml

```

@model project1.Models.signup
@using CaptchaMvc.HtmlHelpers;

@{
    ViewBag.Title = "signup";
}

<script language="javascript" src="http://code.jquery.com/jquery-1.11.1.js"></script>
<script language="javascript" type="text/javascript">
    function AcceptTermAndcondition() {

```

```

        if ($("#chkcondition").is(':checked')) {
            return true;
        }
        else {
            alert("Please accept term and condition");
            return false;
        }
    }
}
</script>

```

```

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>signup</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.name, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.name, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.name, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.empid, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.empid, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.empid, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">

```



```

        @Html.LabelFor(model => model.mobilenno, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.mobilenno, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.mobilenno, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.dob, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.dob, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.dob, "", new { @class = "text-danger"
})
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.emailid, htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.emailid, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.emailid, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.password, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.password, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.password, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">

```

```

        @Html.LabelFor(model => model.confirmpassword, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.confirmpassword, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.confirmpassword, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.securityquestion, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("securityquestion", new List<SelectListItem>
{ new SelectListItem{Text="What is your favorite food?",Value="What is your favorite
food?",Selected=true},
            new SelectListItem{Text="What is your favorite sport?",Value="What is your favorite
sport?"},
            new SelectListItem{Text="What is your favorite cricketer?",Value="Who is your favorite
cricketer?"}});
            @*@Html.EditorFor(model => model.securityquestion, new { htmlAttributes = new {
@class = "form-control" } })*@
            @Html.ValidationMessageFor(model => model.securityquestion, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.securityanswer, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.securityanswer, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.securityanswer, "", new { @class =
"text-danger" })
        </div>
    </div>
</div>

    @Html.TextArea("Termsandcondition", "Term and condition of services.....", new {
@height = "135px", @width = "251px", @id = "txtcondition" })
    </div>@Html.CheckBox("Condition", false, new { @id = "chkcondition" })<a href="#">Term
and condition</a>
<br />

```

```
<div>
    @Html.Captcha(5)

    <p class="Error">@ViewBag.ErrorMessage</p>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="submit" onclick="javascript:return
AcceptTermAndcondition();" class="btn btn-default" />
    </div>
</div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

@model project1.Models.forgot

forgot

```
@using (Html.BeginForm())
```

```

{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>forgot</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.emailid, htmlAttributes: new { @class = "control-label
col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.emailid, new { htmlAttributes = new { @class =
"form-control" } })
                @Html.ValidationMessageFor(model => model.emailid, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.securityquestion, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("securityquestion", new List<SelectListItem>
{ new SelectListItem{Text="What is your favorite food?",Value="What is your favorite
food?",Selected=true},
new SelectListItem{Text="What is your favorite sport?",Value="What is your favorite sport?"},
new SelectListItem{Text="What is your favorite cricketer?",Value="Who is your favorite
cricketer?"}});
                @*@Html.EditorFor(model => model.securityquestion, new { htmlAttributes = new {
@class = "form-control" } })*@
                @Html.ValidationMessageFor(model => model.securityquestion, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.securityanswer, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.securityanswer, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.securityanswer, "", new { @class =
"text-danger" })
            </div>
        </div>
    </div>
}

```

```

</div>

<div class="form-group">
  @Html.LabelFor(model => model.newpassword, htmlAttributes: new { @class =
"control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.newpassword, new { htmlAttributes = new { @class
= "form-control" } })
    @Html.ValidationMessageFor(model => model.newpassword, "", new { @class =
"text-danger" })
  </div>
</div>

<div class="form-group">
  @Html.LabelFor(model => model.confirmpassword, htmlAttributes: new { @class =
"control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.confirmpassword, new { htmlAttributes = new {
@class = "form-control" } })
    @Html.ValidationMessageFor(model => model.confirmpassword, "", new { @class =
"text-danger" })
  </div>
</div>

<div class="form-group">
  <div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Create" class="btn btn-default" />
  </div>
</div>
</div>
}

<div>
  @Html.ActionLink("Back to Login", "Index")
</div>

@section Scripts {
  @Scripts.Render("~/bundles/jqueryval")
}

```

PAGE:Raise query,Leave and Staff manual

1.Sql Tables:

```
create table raise
(id int primary key identity(1,1),
 curloc varchar(250),
 precontim varchar(250),
 shortdes nvarchar(500),
 des nvarchar(1000))
```

```
create table leave(
id int,
typeofleave varchar(250),
startdate date,
enddate date,
comments nvarchar(250))
```

2.vs code

Controller code:

```
using Sampleproject.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Sampleproject.Controllers
{
    public class SampleController : Controller
    { projEntities2 p = new projEntities2();
```

```

// GET: Sample
public ActionResult Index()
{
    return View();
}
[HttpGet]
public FileResult Download()
{
    byte[] fileBytes =
System.IO.File.ReadAllBytes(@"C:\Users\rgnanasekaran\Downloads\
staff manual.pdf");
    string filename = "staff manual.pdf";
    return File(fileBytes,
System.Net.Mime.MediaTypeNames.Application.Octet, filename);
}
[HttpGet]
public ActionResult leave()
{
    return View();
}
[HttpPost]
public ActionResult leave(leave m)
{
    p.leaves.Add(m);
    p.SaveChanges();
    ViewBag.Message = "Data insert successfully";
    return View();
}
[HttpGet]
public ActionResult raise()
{
    return View();
}

```

```

    }
    [HttpPost]
    public ActionResult raise(raise r)
    {
        p.raises.Add(r);
        p.SaveChanges();
        ViewBag.Message = "Data insert successfully";
        return View();
    }
}

```

Index view(Staff manual):

```

@{
    ViewBag.Title = "Index";
}

```

<h2>Index</h2>

@Html.ActionLink("Download Staff Manual", "Download", "Home")

Leave view:

@model Sampleproject.Models.leave

```

@{
    ViewBag.Title = "leave";
}

```

<h2>leave</h2>


```
<link  
href="https://ajax.aspnetcdn.com/ajax/jquery.ui/1.10.4/themes/humanit  
y/jquery-ui.css" rel="stylesheet" />
```

```
@section Scripts{  
    <script src="~/Scripts/jquery-3.4.1.min.js"></script>  
    <script src="~/Scripts/jquery-ui-1.12.1.min.js"></script>  
    <link href="~/Content/themes/base/selectable.css" rel="stylesheet"  
/>
```

```
    <script>  
        $(document).ready(function () {  
            $("#startdate").datepicker();  
            $("#enddate").datepicker();  
        })  
    </script>
```

```
}
```

```
@using (Html.BeginForm())
```

```
{  
    @Html.AntiForgeryToken()  
  
    <div class="form-horizontal">  
        <h4>leave</h4>  
        <hr />  
        @Html.ValidationSummary(true, "", new { @class = "text-danger"  
    })
```

```
        <div class="form-group">  
            @Html.LabelFor(model => model.typeofleave, "Type of leave",  
htmlAttributes: new { @class = "control-label col-md-2" })  
            <div class="col-md-10">  
                @Html.DropDownList("typeofleave", new  
List<SelectListItem>
```

```

        {new
SelectListItem{Text="---Type---",Value="---type---",Selected=true},
        new SelectListItem{Text="Compensatory
leave",Value="Compensatory Leave" },
        new SelectListItem{Text="Flexible Statutory
leave",Value="Flexible Statutory Leave" },
        new SelectListItem{Text="Annual leave",Value="Annual
Leave" },
        new SelectListItem{Text="Compassionate
leave",Value="Compassionate Leave" },
        new SelectListItem{Text="Emergency Medical
leave",Value="Emergency medical Leave" },
        new SelectListItem{Text="Sick Leave",Value="Sick Leave" },
        new SelectListItem{Text="Paternity leave",Value="Paternity
Leave" },
        new SelectListItem{Text="Maternity
leave",Value="Maternity Leave" },
        });
    </div>
</div>

```

```

<div class="form-group">
    @Html.LabelFor(model => model.startdate,"Start Date",
htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.startdate, new {
htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.startdate, "",
new { @class = "text-danger" })
    </div>
</div>

```

```

        <div class="form-group">
            @Html.LabelFor(model => model.enddate,"End date",
htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.enddate, new {
htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.enddate, "",
new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.comments,"Comments",
htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.comments, new {
htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.comments,
"", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default"
/>
            </div>
        </div>
    </div>
}

<div>

```

```
    @Html.ActionLink("Back to List", "Index")
</div>
```

Raise query view:

```
@model Sampleproject.Models.raise
```

```
@{
    ViewBag.Title = "raise";
}
```

```
<h2>raise</h2>
```

```
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>raise</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger"
    })
        <div class="form-group">
            @Html.LabelFor(model => model.curloc, "Current Location",
htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("curloc", new List<SelectListItem>
                {new
SelectListItem{Text="---Type---",Value="---type---",Selected=true},
                new SelectListItem{Text="Base office",Value="Base office" },
                new SelectListItem{Text="other KGS office",Value="Other
KGS office" },
```

```

        new SelectListItem{Text="Work from home",Value="Work
from home" },
        new SelectListItem{Text="Out of country",Value="out of
country" },
        new SelectListItem{Text="Client location",Value="Client
location" },
    });
</div>
</div>

```

```

<div class="form-group">
    @Html.LabelFor(model => model.precontim, "Preferred
contact timing", htmlAttributes: new { @class = "control-label
col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("precontim",new List<SelectListItem>
        {new
SelectListItem{Text="---Type---",Value="---type---",Selected=true},
        new SelectListItem{Text="any time",Value="Any time" },
        new SelectListItem{Text="08.00-12.00
hrs",Value="08.00-12.00 hrs" },
        new SelectListItem{Text="12.00-16.00
hrs",Value="12.00-16.00 hrs" },
        new SelectListItem{Text="16.00-20.00
hrs",Value="16.00-20.00 hrs" },
        new SelectListItem{Text="20.00-00.00
hrs",Value="20.00-00.00 hrs" },
        new SelectListItem{Text="00.00-04.00
hrs",Value="00.00-04.00 hrs" },
        new SelectListItem{Text="04.00-08.00
hrs",Value="04.00-08.00 hrs" },
        });
    </div>
</div>

```

```
</div>
</div>
```

```
<div class="form-group">
  @Html.LabelFor(model => model.shortdes,"Short description",
htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.shortdes, new {
htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.shortdes, "",
new { @class = "text-danger" })
  </div>
</div>
```

```
<div class="form-group">
  @Html.LabelFor(model => model.des,"Description",
htmlAttributes: new { @class = "control-label col-md-2" })
  <div class="col-md-10">
    @Html.EditorFor(model => model.des, new { htmlAttributes
= new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.des, "", new
{ @class = "text-danger" })
  </div>
</div>
```

```
<div class="form-group">
  <div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Create" class="btn btn-default"
/>
  </div>
</div>
</div>
```

```
}
```

```
<div>
```

```
    @Html.ActionLink("Back to List", "Index")
```

```
</div>
```

PAGE:SALARY DETAILS

Salary.cs

```
namespace Proj.Models
{
    public class Salary
    {
        public double basicpay { get; set; }
        public double da { get; set; }
        public double hra { get; set; }
        public double pf { get; set; }
        public double tax { get; set; }
        public double netsalary { get; set; }
    }
    public class SalaryModel
    {
        public List<Salary> SalaryList { get; set; }
    }
}
```

Salarycontroller.cs

```
namespace Proj.Controllers
{
    public class SalaryController : Controller
    {
        // GET: Salary
        public ActionResult Index()
        {
            return View(new Salary());
        }
        [HttpPost]
        public ActionResult Index(Salary s)
        {
            if (s.basicpay > 60000)
            {
                s.da = s.basicpay * 5 / 100;
                s.hra = s.basicpay * 35 / 100;
                s.pf = s.basicpay * 5 / 100;
                s.tax = s.basicpay * 10 / 100;
            }
        }
    }
}
```



```

    }
    else if (s.basicpay > 30000)
    {
        s.da = s.basicpay * 5 / 100;
        s.hra = s.basicpay * 25 / 100;
        s.pf = s.basicpay * 5 / 100;
        s.tax = s.basicpay * 10 / 100;
    }
    else
    {
        s.da = s.basicpay * 5 / 100;
        s.hra = s.basicpay * 15 / 100;
        s.pf = s.basicpay * 2.5 / 100;
        s.tax = s.basicpay * 0 / 100;
    }

    s.netsalary = s.basicpay + s.da + s.hra - s.pf - s.tax;

    //List<Salary> _salary = new List<Salary>();
    //_salary.Add(new Salary { basicpay = s.basicpay, da = s.da, hra = s.hra, pf = s.pf, tax =
s.tax, netsalary = s.netsalary });

    //SalaryModel _salaryModel = new SalaryModel();
    //_salaryModel.SalaryList = _salary;
    return View(s);
}

}
}

```

Index.cshtml

@model Proj.Models.Salary

```
@{
    ViewBag.Title = "Index";
}
<style type="text/css">
    .gridtable {
        font-family: verdana,arial,sans-serif;
        font-size: 11px;
        color: #333333;
        border-width: 1px;
        border-color: #666666;
        border-collapse: collapse;
    }

    .gridtable th {
        border-width: 1px;
        padding: 8px;
        border-style: solid;
        border-color: #666666;
        background-color: #dedede;
    }

    .gridtable td {
        border-width: 1px;
        padding: 8px;
        border-style: solid;
        border-color: #666666;
        background-color: #ffffff;
    }

    .gridtable tr:hover td,
    .gridtable tr.even:hover td.active,
    .gridtable tr.odd:hover td.active {
        background: #b6ff00;
    }
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
.center {
    border: 5px solid #FFFF00;
    text-align: center;
```

```

    }
</style>

<h2>Index</h2>

<html>
<head>
    Salary Calculator
</head>
<body>
    <div>
        @using (Html.BeginForm("/Index", "Salary", FormMethod.Post))
        {
            <div class="form-group">
                <label for="name">Basic Pay</label>
                <input type="text" class="form-control" id="basicpay" name="basicpay"
value="@Model.basicpay" />
            </div>
            <div class="form-group">
                <label for="name">HRA</label>
                <input type="text" class="form-control" id="hra" name="hra" value="@Model.hra" />
            </div>
            <div class="form-group">
                <label for="name">DA</label>
                <input type="text" class="form-control" id="da" name="da" value="@Model.da" />
            </div>
            <div class="form-group">
                <label for="name">PF</label>
                <input type="text" class="form-control" id="pf" name="pf" value="@Model.pf" />
            </div>
            <div class="form-group">
                <label for="name">Tax</label>
                <input type="text" class="form-control" id="tax" name="tax" value="@Model.tax" />
            </div>
            <div class="form-group">
                <label for="name">Net Salary</label>
                <input type="text" class="form-control" id="netsalary" name="netsalary"
value="@Model.netsalary" />
            </div>
            <button type="submit" class="btn btn-info" id="add" value="add">Calculate</button>
        }
    </div>

</div>

```

```
<table>
  <tr>
    <th>Employee Name</th>
    <td>Name</td>
  </tr>
  <tr>
    <th>Basic Pay </th>
    <td>@Model.basicpay</td>
  </tr>
  <tr>
    <th>DA    </th>
    <td>@Model.da</td>
  </tr>
  <tr>
    <th>HRA    </th>
    <td>@Model.hra</td>
  </tr>
  <tr>
    <th>PF    </th>
    <td>@Model.pf</td>
  </tr>
  <tr>
    <th>Tax    </th>
    <td>@Model.tax</td>
  </tr>
  <tr>
    <th>Net Salary</th>
    <td>@Model.netsalary</td>
  </tr>
</table>
</div>
</body>
</html>
```

