private void Function\_Name(long Page = 1)

{

string qry = "";

DataSet ds = dl.Inline\_Process(qry, "CreatedOn", "DESC", Page, pagesize);

DataTable dt = new DataTable();

if (ds != null && ds.Tables.Count > 0)

{

dt = ds.Tables[1];

}

long Tot = 0, totalrows = 0;

if (dt.Rows.Count > 0)

{

long.TryParse(dt.Rows[0]["TotalPage"].ToString(), out Tot);

long.TryParse(dt.Rows[0]["Total"].ToString(), out totalrows);

}

ViewBag.TotalPage = Tot;

ViewBag.CurrentPage = Page;

ViewBag.Propertylst = ds;

var pager = new Pager(totalrows, Page, pagesize);

ViewBag.PageNoList = pager;

}

**View**

Pager PModel = (Pager)ViewBag.PageNoList;

long TotalPage = (long)ViewBag.TotalPage;

long CurrPage = (long)ViewBag.CurrentPage;

<div class="pagination">

<div class="center">

@if (PModel.EndPage > 1)

{

<ul>

@if (PModel.CurrentPage > 1)

{

<li>

<a href="@Url.Action("Index", "Property")" class="button small grey">First</a>

</li>

<li>

<a href="@Url.Action("Index", "Property", new { @page = (PModel.CurrentPage - 1) })">Previous</a>

</li>

}

@for (var page = PModel.StartPage; page <= PModel.EndPage; page++)

{

<li class="@(page == PModel.CurrentPage ? "active" : "")">

<a href="@Url.Action("Index", "Property", new { @page = @page })">@page</a>

</li>

}

@if (PModel.CurrentPage < PModel.TotalPages)

{

<li>

<a href="@Url.Action("Index", "Property", new { @page = (PModel.CurrentPage + 1) })">Next</a>

</li>

<li>

<a href="@Url.Action("Index", "Property", new { @page = (PModel.TotalPages) })">Last</a>

</li>

}

</ul>

}

</div>

<div class="clear"></div>

</div>

//Page wise Inline query-------------

public DataSet Inline\_Process(String Query, string OrderBy, string AscDesc, long Page, long PageSize)

{

string[] paraname = {"@Qry", "@OrderBy", "@ASCDESC", "@Page", "@rowsPerPage"};

string[] paravalue = {Query, OrderBy, AscDesc, Page.ToString(), PageSize.ToString()};

DataSet ds = Executeproc("ExecuteQueryPageWise", paraname, paravalue);

return ds;

}

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ExecuteQueryPageWise] Script Date: 09-04-2018 9.11.01 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create Procedure [dbo].[ExecuteQueryPageWise]

(

@Qry nvarchar(max)='',

@OrderBy nvarchar(Max),

@ASCDESC varchar(20)='DESC',

@Page bigint=1,

@rowsPerPage bigint=10

)

as

begin

declare @SQLQuery AS NVARCHAR(MAX)

declare @TOTALPage as bigint;

declare @TOTAL as bigint;

declare @ParaDefination nvarchar(Max)

declare @pageNum as bigint;

set @pageNum=@Page;

--select @TOTAL=COUNT(Restaurant\_Id) from view\_RestaurantList where Status=1 and Restaurant\_Type=@Restaurant\_Type and (Address like '%'+@Address+'%' or City like '%'+@Address+'%' or State like '%' + @Address + '%' or Country like '%' + @Address + '%');

set @ParaDefination='@TOTAL bigint=0 output'

--Set @SQLQuery='select @TOTAL=COUNT(Restaurant\_Id) from view\_RestaurantList where Status=1 and Restaurant\_Type=@Restaurant\_Type and (Address like ''%'+@Address+'%'' or City like ''%'+@Address+'%'' or State like ''%' + @Address + '%'' or Country like ''%' + @Address + '%'')'+ @Filter;

set @SQLQuery ='Select @TOTAL=Count(\*) from ('+@Qry+') A'

print @SQLQuery

EXECUTE sp\_executesql @SQLQuery,@ParaDefination,@TOTAL output;

print @TOTAL

Set @SQLQuery='With SQLPaging As (

Select Top(@rowsPerPage \* @pageNum) ROW\_NUMBER() OVER (ORDER BY '+@OrderBy+' '+@ASCDESC+')

as RowNum, \*

FROM ('+@Qry+') A'

set @SQLQuery=@SQLQuery+') select \* from SQLPaging with (nolock) where RowNum > ((@pageNum - 1) \* @rowsPerPage) order by RowNum ASC'

set @ParaDefination='@rowsPerPage bigint,@pageNum bigint'

EXECUTE sp\_executesql @SQLQuery,@ParaDefination,@rowsPerPage,@pageNum;

---------------------------------------------Calculate Pages-------------------------

set @TOTALPage=@TOTAL%@rowsPerPage

if(@TOTALPage=0)

begin

set @TOTALPage=@TOTAL/@rowsPerPage

end

else

begin

set @TOTALPage=(@TOTAL/@rowsPerPage)+1

end

SELECT @TOTAL as Total,@TOTALPage as TotalPage

--------------------------------------------------------------------------------------

end

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ExecuteQueryPageWise\_v3] Script Date: 09-04-2018 9.11.08 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create Procedure [dbo].[ExecuteQueryPageWise\_v3]

(

@Qry nvarchar(max)='',

@OrderBy nvarchar(Max),

@Page bigint=1,

@rowsPerPage bigint=10

)

as

begin

declare @SQLQuery AS NVARCHAR(MAX)

declare @TOTALPage as bigint;

declare @TOTAL as bigint;

declare @ParaDefination nvarchar(Max)

declare @pageNum as bigint;

set @pageNum=@Page;

--select @TOTAL=COUNT(Restaurant\_Id) from view\_RestaurantList where Status=1 and Restaurant\_Type=@Restaurant\_Type and (Address like '%'+@Address+'%' or City like '%'+@Address+'%' or State like '%' + @Address + '%' or Country like '%' + @Address + '%');

set @ParaDefination='@TOTAL bigint=0 output'

--Set @SQLQuery='select @TOTAL=COUNT(Restaurant\_Id) from view\_RestaurantList where Status=1 and Restaurant\_Type=@Restaurant\_Type and (Address like ''%'+@Address+'%'' or City like ''%'+@Address+'%'' or State like ''%' + @Address + '%'' or Country like ''%' + @Address + '%'')'+ @Filter;

set @SQLQuery ='Select @TOTAL=Count(\*) from ('+@Qry+') A'

print @SQLQuery

EXECUTE sp\_executesql @SQLQuery,@ParaDefination,@TOTAL output;

print @TOTAL

Set @SQLQuery='With SQLPaging As (

Select Top(@rowsPerPage \* @pageNum) ROW\_NUMBER() OVER (ORDER BY '+@OrderBy+')

as RowNum, \*

FROM ('+@Qry+') A'

set @SQLQuery=@SQLQuery+') select \* from SQLPaging with (nolock) where RowNum > ((@pageNum - 1) \* @rowsPerPage) order by RowNum ASC'

set @ParaDefination='@rowsPerPage bigint,@pageNum bigint'

EXECUTE sp\_executesql @SQLQuery,@ParaDefination,@rowsPerPage,@pageNum;

---------------------------------------------Calculate Pages-------------------------

set @TOTALPage=@TOTAL%@rowsPerPage

if(@TOTALPage=0)

begin

set @TOTALPage=@TOTAL/@rowsPerPage

end

else

begin

set @TOTALPage=(@TOTAL/@rowsPerPage)+1

end

SELECT @TOTAL as Total,@TOTALPage as TotalPage

--------------------------------------------------------------------------------------

end

GO