HomeController

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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace NIMAP\_TASK.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

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

}

}

}

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

ProductController

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

using NIMAP\_TASK.Models;

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data.Entity;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace NIMAP\_TASK.Controllers

{

public class ProductController : Controller

{

public ActionResult Index(int page = 1, int pageSize = 10)

{

List<ProductModel> products = GetProducts(page, pageSize);

ViewBag.CurrentPage = page;

ViewBag.PageSize = pageSize;

ViewBag.TotalProducts = GetTotalProductsCount();

return View(products);

}

public List<ProductModel> GetProducts(int page, int pageSize)

{

List<ProductModel> products = new List<ProductModel>();

int skip = (page - 1) \* pageSize;

string con = ConfigurationManager.ConnectionStrings["Test"].ConnectionString;

using (SqlConnection connection = new SqlConnection(con))

{

string query = "SELECT TOP(@PageSize) \* FROM (SELECT ROW\_NUMBER() OVER (ORDER BY ProductId) AS RowNum, " +

"ProductId, ProductName, CategoryId, CategoryName FROM Products) AS P WHERE RowNum > @Skip";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@PageSize", pageSize);

command.Parameters.AddWithValue("@Skip", skip);

connection.Open();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

ProductModel product = new ProductModel

{

ProductId = reader.GetInt32(1),

ProductName = reader.GetString(2),

CategoryId = reader.GetInt32(3),

CategoryName = reader.GetString(4)

};

products.Add(product);

}

}

}

return products;

}

public int GetTotalProductsCount()

{

int totalProducts = 0;

string con = ConfigurationManager.ConnectionStrings["Test"].ConnectionString;

using (SqlConnection connection = new SqlConnection(con))

{

string query = "SELECT COUNT(\*) FROM Products";

using (SqlCommand command = new SqlCommand(query, connection))

{

connection.Open();

totalProducts = (int)command.ExecuteScalar();

}

}

return totalProducts;

}

}

}

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

CategoryController

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

using NIMAP\_TASK.Models;

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace NIMAP\_TASK.Controllers

{

public class CategoryController : Controller

{

public ActionResult Index()

{

List<CategoryModel> categories = GetCategoriesFromDatabase();

return View(categories);

}

private List<CategoryModel> GetCategoriesFromDatabase()

{

List<CategoryModel> categories = new List<CategoryModel>();

string con = ConfigurationManager.ConnectionStrings["Test"].ConnectionString;

using (var connection = new SqlConnection(con))

{

connection.Open();

using (var command = new SqlCommand("SELECT \* FROM Categories", connection))

{

using (var reader = command.ExecuteReader())

{

while (reader.Read())

{

CategoryModel category = new CategoryModel

{

CategoryId = reader.GetInt32(0),

CategoryName = reader.GetString(1)

};

categories.Add(category);

}

}

}

}

return categories;

}

}

}

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

Index

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

@{

ViewBag.Title = "Index";

}

@model List<NIMAP\_TASK.Models.ProductModel>

<html>

<head>

<style>

.tbl

{

margin-left:35%;

}

table th {

background-color: #A4C2F4;

border: 1px solid #ddd;

padding: 8px;

}

a {

text-decoration: none;

border: 1px solid black;

padding:7px;

background-color: #A4C2F4;

}

.p {

margin-left: 45%;

}

td

{

border: 1px solid #ddd;

padding: 8px;

text-align:center;

}

</style>

</head>

<body>

<table class="table tbl">

<tr>

<th>ProductId</th>

<th>ProductName</th>

<th>CategoryId</th>

<th>CategoryName</th>

</tr>

@foreach (var product in Model)

{

<tr>

<td>@product.ProductId</td>

<td>@product.ProductName</td>

<td>@product.CategoryId</td>

<td>@product.CategoryName</td>

</tr>

}

</table>

@{

int totalPages = (int)Math.Ceiling((double)ViewBag.TotalProducts / ViewBag.PageSize);

}

<br />

@if (totalPages > 1)

{

<div class="p">

@for (int i = 1; i <= totalPages; i++)

{

if (i == ViewBag.CurrentPage)

{

<a>@i</a>

}

else

{

<a href="@Url.Action("Index", new { page = i, pageSize = ViewBag.PageSize })">@i</a>

}

}

</div>

}

</body>

</html>

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

CategoryModel

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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace NIMAP\_TASK.Models

{

public class CategoryModel

{

public int CategoryId { get; set; }

public string CategoryName { get; set; }

}

}

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

ProductModel

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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace NIMAP\_TASK.Models

{

public class ProductModel

{

public int ProductId { get; set; }

public string ProductName { get; set; }

public int CategoryId { get; set; }

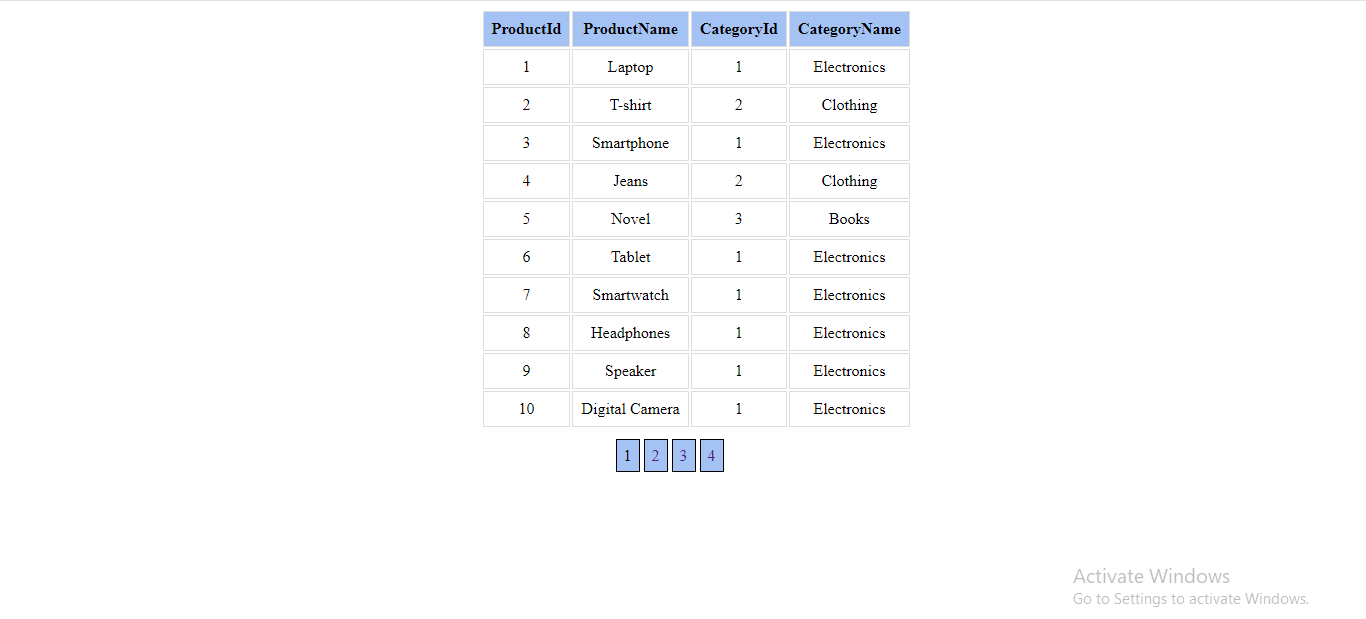
public string CategoryName { get; set; }

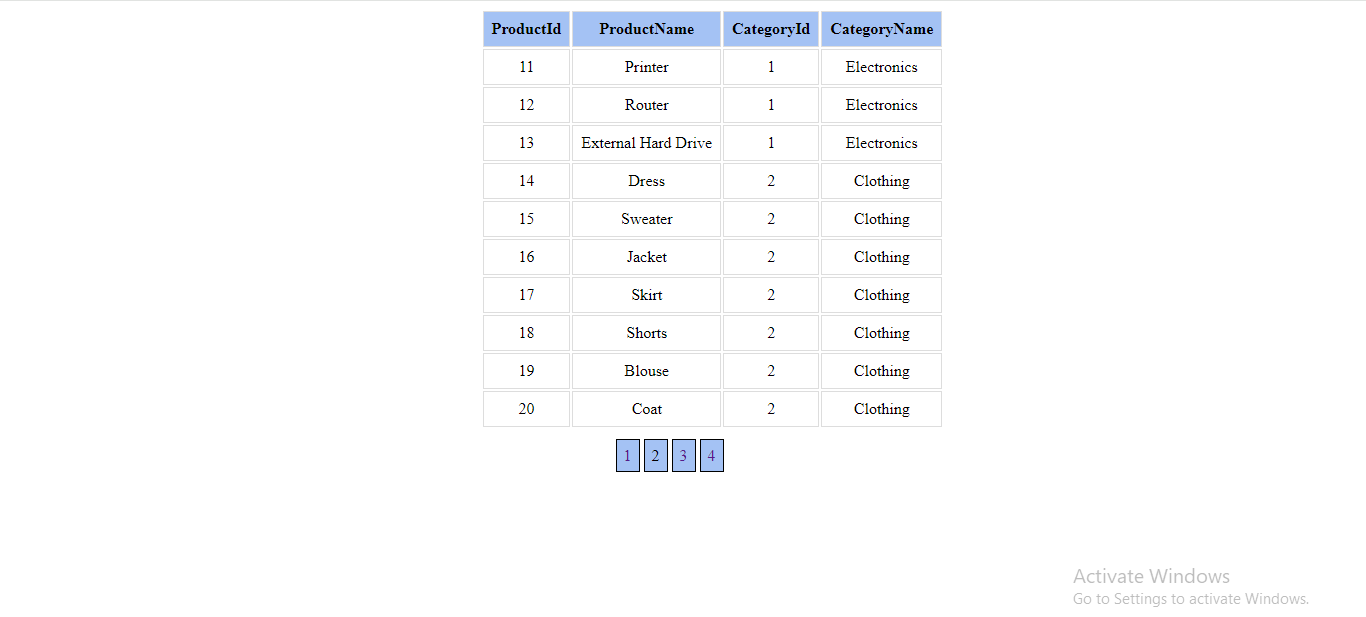
}

}

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

Output







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