🡪Controller

using TableDemo.Models;

using Microsoft.AspNetCore.Mvc;

using System.Data;

using System.Data.SqlClient;

using OfficeOpenXml;

using TableDemo.Models;

using TableDemo;

using TableDemo.Helper;

namespace addressbook\_app.Controllers

{

[CheckAccess]

public class CountryController : Controller

{

private IConfiguration \_configuration;

public CountryController(IConfiguration conf)

{

\_configuration = conf;

}

#region CountryList

public IActionResult CountryList()

{

string connectionString = this.\_configuration.GetConnectionString("ConnectionString");

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlCommand command = connection.CreateCommand();

command.CommandType = System.Data.CommandType.StoredProcedure;

command.CommandText = "PR\_Country\_SelectAll";

SqlDataReader reader = command.ExecuteReader();

DataTable dataTable = new DataTable();

dataTable.Load(reader);

return View(dataTable);

}

#endregion

#region CountryDelete

public IActionResult CountryDelete(string countryID)

{

int decryptedCountryID = Convert.ToInt32(UrlEncryptor.Decrypt(countryID.ToString()));

try

{

string connectionString = this.\_configuration.GetConnectionString("ConnectionString");

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlCommand command = connection.CreateCommand();

command.CommandType = System.Data.CommandType.StoredProcedure;

command.CommandText = "PR\_Country\_DeleteByPK";

//command.Parameters.Add("@CountryID", SqlDbType.Int).Value = countryID;

command.Parameters.AddWithValue("@CountryID", decryptedCountryID);

command.ExecuteNonQuery();

}

catch (Exception ex)

{

TempData["ErrorMsg"] = ex.Message;

}

return RedirectToAction("CountryList");

}

#endregion

#region CountryForm

public IActionResult CountryForm(string? countryID)

{

int? decryptedCountryID = null;

// Decrypt only if CityID is not null or empty

if (!string.IsNullOrEmpty(countryID))

{

string decryptedCountryIDString = UrlEncryptor.Decrypt(countryID); // Decrypt the encrypted CityID

decryptedCountryID = int.Parse(decryptedCountryIDString); // Convert decrypted string to integer

}

if (countryID == null)

{

var m = new CountryModel

{

CreatedDate = DateTime.Now

};

return View(m);

}

string connectionString = this.\_configuration.GetConnectionString("ConnectionString");

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlCommand command = connection.CreateCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "PR\_Country\_SelectByPK";

//command.Parameters.AddWithValue("@CountryID", countryID);

command.Parameters.Add("@CountryID", SqlDbType.Int).Value = decryptedCountryID;

SqlDataReader reader = command.ExecuteReader();

DataTable table = new DataTable();

table.Load(reader);

CountryModel model = new CountryModel();

foreach (DataRow dataRow in table.Rows)

{

model.CountryName = dataRow["CountryName"].ToString();

model.CountryCode = dataRow["CountryCode"].ToString();

}

return View(model);

}

#endregion

//#region CountryAddEdit

//public IActionResult CountryAddEdit(CountryModel model)

//{

// if (ModelState.IsValid)

// {

// string connectionString = this.\_configuration.GetConnectionString("ConnectionString");

// SqlConnection connection = new SqlConnection(connectionString);

// connection.Open();

// SqlCommand command = connection.CreateCommand();

// command.CommandType = CommandType.StoredProcedure;

// if (model.CountryID == 0)

// {

// command.CommandText = "PR\_Country\_Insert";

// }

// else

// {

// command.CommandText = "PR\_Country\_UpdateByPK";

// command.Parameters.Add("@CountryID", SqlDbType.Int).Value = model.CountryID;

// }

// command.Parameters.Add("@CountryName", SqlDbType.VarChar).Value = model.CountryName;

// command.Parameters.Add("@CountryCode", SqlDbType.VarChar).Value = model.CountryCode;

// command.ExecuteNonQuery();

// TempData["SuccessMsg"] = "Task performed successfully";

// return RedirectToAction("CountryList");

// }

// return View("CountryForm", model);

//}

//#endregion

[HttpPost]

public IActionResult CountryAddEdit(CountryModel model)

{

if (ModelState.IsValid)

{

try

{

string connectionString = this.\_configuration.GetConnectionString("ConnectionString");

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

using (SqlCommand command = connection.CreateCommand())

{

command.CommandType = CommandType.StoredProcedure;

if (model.CountryID == 0)

{

command.CommandText = "PR\_Country\_Insert";

}

else

{

command.CommandText = "PR\_Country\_UpdateByPK";

command.Parameters.Add("@CountryID", SqlDbType.Int).Value = model.CountryID;

}

command.Parameters.Add("@CountryName", SqlDbType.VarChar).Value = model.CountryName;

command.Parameters.Add("@CountryCode", SqlDbType.VarChar).Value = model.CountryCode;

command.ExecuteNonQuery();

}

}

return Json(new { success = true, message = "Country saved successfully!" });

}

catch (Exception ex)

{

return Json(new { success = false, message = ex.Message });

}

}

return Json(new { success = false, message = "Invalid data." });

}

#region Export to excel

public IActionResult ExportToExcel()

{

ExcelPackage.LicenseContext = LicenseContext.NonCommercial;

string connectionString = \_configuration.GetConnectionString("ConnectionString");

SqlConnection sqlConnection = new SqlConnection(connectionString);

sqlConnection.Open();

SqlCommand sqlCommand = sqlConnection.CreateCommand();

sqlCommand.CommandType = System.Data.CommandType.StoredProcedure;

sqlCommand.CommandText = "PR\_Country\_SelectAll";

SqlDataReader sqlDataReader = sqlCommand.ExecuteReader();

DataTable data = new DataTable();

data.Load(sqlDataReader);

using (var package = new ExcelPackage())

{

var worksheet = package.Workbook.Worksheets.Add("DataSheet");

// Add headers

worksheet.Cells[1, 1].Value = "CountryID";

worksheet.Cells[1, 2].Value = "CountryName";

worksheet.Cells[1, 3].Value = "CountryCode";

worksheet.Cells[1, 5].Value = "CreatedDate";

worksheet.Cells[1, 5].Value = "ModifiedDate";

// Add data

int row = 2;

foreach (DataRow item in data.Rows)

{

worksheet.Cells[row, 1].Value = item["CountryID"];

worksheet.Cells[row, 2].Value = item["CountryName"];

worksheet.Cells[row, 3].Value = item["CountryCode"];

worksheet.Cells[row, 4].Value = item["CreatedDate"];

worksheet.Cells[row, 5].Value = item["ModifiedDate"];

row++;

}

var stream = new MemoryStream();

package.SaveAs(stream);

stream.Position = 0;

string excelName = $"CountryData.xlsx";

return File(stream, "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet", excelName);

}

#endregion

}

}

}

---AddEdit Page

@model TableDemo.Models.CountryModel

<div class="container mt-5">

<div class="page-bar mb-4">

<ul class="page-breadcrumb list-inline">

<li class="list-inline-item">

<i class="icon-home"></i>

<**a** **asp-controller**="Home" **asp-action**="Index">Home</**a**>

<i class="fa fa-arrow-right"></i>

</li>

<li class="list-inline-item">

<span>Country Add/Edit</span>

</li>

</ul>

</div>

<div class="card mx-auto" style="max-width: 900px;">

<div class="card-header text-center">

<h4 class="font-weight-bold">Country Add/Edit</h4>

</div>

<div class="card-body">

<**form** id="countryForm" class="form-horizontal">

@Html.HiddenFor(m => m.CountryID)

<div class="form-group row mb-3">

<label class="col-md-3 col-form-label text-md-end"><span class="text-danger">\*</span>Country Name</label>

<div class="col-md-9">

<input type="text" class="form-control" id="CountryName" name="CountryName" value="@Model.CountryName" placeholder="Enter Country Name">

<**span** **asp-validation-for**="CountryName" class="text-danger"></**span**>

</div>

</div>

<div class="form-group row mb-3">

<label class="col-md-3 col-form-label text-md-end"><span class="text-danger">\*</span>Country Code</label>

<div class="col-md-9">

<input type="text" class="form-control" id="CountryCode" name="CountryCode" value="@Model.CountryCode" placeholder="Enter Country Code">

<**span** **asp-validation-for**="CountryCode" class="text-danger"></**span**>

</div>

</div>

<div class="form-actions text-center">

<button type="submit" class="btn btn-secondary px-4">Submit</button>

<**a** **asp-controller**="Country" **asp-action**="CountryList" class="btn btn-secondary px-4">Cancel</**a**>

</div>

</**form**>

</div>

</div>

</div>

@section Scripts {

<script>

$(document).ready(function () {

$("#countryForm").on("submit", function (e) {

e.preventDefault();

var formData = {

CountryID: $("#CountryID").val(),

CountryName: $("#CountryName").val(),

CountryCode: $("#CountryCode").val()

};

$.ajax({

url: '@Url.Action("CountryAddEdit", "Country")',

type: 'POST',

data: formData,

success: function (response) {

if (response.success) {

alert("Country saved successfully!");

window.location.href = '@Url.Action("CountryList", "Country")';

} else {

alert(response.message);

}

},

error: function () {

alert("Failed to save the country. Please try again.");

}

});

});

});

</script>

}

---List PAGE

@using System.Data;

@using TableDemo.Helper

@model DataTable;

<main id="main" class="main">

<h1>Country List</h1>

<div class="d-flex justify-content-end mb-3">

<div class="me-2">

<**a** **asp-controller**="Country" **asp-action**="CountryForm">

<button type="button" class="btn btn-outline-primary">Add New Country</button>

</**a**>

</div>

<div class="me-2">

<**a** **asp-controller**="Country" **asp-action**="ExportToExcel">

<button type="button" class="btn btn-outline-secondary">Export to Excel</button>

</**a**>

</div>

</div>

<div class="row justify-content-end" style="padding-bottom:10px">

<div class="col-3">

<input type="text" class="form-control" id="countrySearch" placeholder="Search ...">

</div>

</div>

<span class="text-danger">@TempData["ErrorMsg"]</span>

<span class="text-success">@TempData["SuccessMsg"]</span>

<table class="table table-bordered">

<thead class="text-dark fw-bold text-center">

<tr>

<td>Country Name</td>

<td>Country Code</td>

<td>Actions</td>

</tr>

</thead>

<tbody id="countryTable" class="text-center">

@foreach (DataRow dataRow in Model.Rows)

{

<tr>

<td>@dataRow["CountryName"]</td>

<td>@dataRow["CountryCode"]</td>

<td>

<**a** class="btn btn-outline-success btn-xs" **asp-controller**="Country" **asp-action**="CountryForm" **asp-route-CountryID**="@dataRow["CountryID"]">

<i class="bi bi-pencil-fill"></i>

</**a**>

<**form** method="post" **asp-controller**="Country" **asp-action**="CountryDelete" **asp-route-CountryID**="@dataRow["CountryID"]" onclick="return confirm('Are you sure you want to delete this Country?');" style="display:inline;">

<button type="submit" class="btn btn-outline-danger btn-xs">

<i class="bi bi-x"></i>

</button>

</**form**>

</td>

</tr>

}

</tbody>

</table>

</main>

@section Scripts {

<script>

$(document).ready(function () {

// Search functionality

$("#countrySearch").on("keyup", function () {

var value = $(this).val().toLowerCase();

$("#countryTable tr").filter(function () {

$(this).toggle($(this).text().toLowerCase().indexOf(value) > -1);

});

});

});

</script>

}