# Data access

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Web.Configuration;

using Web\_API.Models;

namespace Web\_API.Data\_Layer

{

public class DataLayer

{

string \_connectionString = "";

public DataLayer()

{

\_connectionString = WebConfigurationManager.ConnectionStrings["DemoCN"].ConnectionString;

}

public DataTable GetListOfSpecific(string param)

{

try

{

List<Posts> lstPosts = new List<Posts>();

DataTable dataTable = new DataTable();

using (SqlConnection con = new SqlConnection(\_connectionString))

{

SqlCommand command = new SqlCommand("GetSpecificPosts", con);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@email", param);

con.Open();

SqlDataAdapter dataAdapter = new SqlDataAdapter(command);

dataAdapter.Fill(dataTable);

con.Close();

}

return dataTable;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in GetListOfStudents due to "

+ exception.Message, exception.InnerException);

}

}

public DataTable GetListOfPosts()

{

try

{

List<Posts> lstPosts = new List<Posts>();

DataTable dataTable = new DataTable();

using (SqlConnection con = new SqlConnection(\_connectionString))

{

SqlCommand command = new SqlCommand("GetPostsList", con);

command.CommandType = CommandType.StoredProcedure;

con.Open();

SqlDataAdapter dataAdapter = new SqlDataAdapter(command);

dataAdapter.Fill(dataTable);

con.Close();

}

return dataTable;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in GetListOfStudents due to "

+ exception.Message, exception.InnerException);

}

}

public string InsertUserInDB(Users user)

{

try

{

string response = "";

using (SqlConnection con = new SqlConnection(\_connectionString))

{

SqlCommand command = new SqlCommand("AddNewUser", con);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@email", user.Email);

command.Parameters.AddWithValue("@name", user.Name);

SqlParameter outputParameter = new SqlParameter("@Message", SqlDbType.VarChar, 100);

outputParameter.Direction = ParameterDirection.Output;

command.Parameters.Add(outputParameter);

con.Open();

command.ExecuteNonQuery();

response = command.Parameters["@Message"].Value.ToString();

con.Close();

}

return response;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in InsertStudentInDB due to "

+ exception.Message, exception.InnerException);

}

}

public string InsertPostInDB(Posts post)

{

try

{

string response = "";

using (SqlConnection con = new SqlConnection(\_connectionString))

{

SqlCommand command = new SqlCommand("InsertPost", con);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@name", post.Name);

command.Parameters.AddWithValue("@email", post.Email);

command.Parameters.AddWithValue("@category", post.Category);

command.Parameters.AddWithValue("@title", post.Title);

command.Parameters.AddWithValue("@content", post.Content);

SqlParameter outputParameter = new SqlParameter("@Message", SqlDbType.VarChar, 100);

outputParameter.Direction = ParameterDirection.Output;

command.Parameters.Add(outputParameter);

con.Open();

command.ExecuteNonQuery();

response = command.Parameters["@Message"].Value.ToString();

con.Close();

}

return response;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in InsertStudentInDB due to "

+ exception.Message, exception.InnerException);

}

}

}

}

# Business layer

using System;

using System.Collections.Generic;

using System.Data;

using Web\_API.Data\_Layer;

using Web\_API.Models;

namespace Web\_API.BusinessLayer

{

public class BL

{

public DataLayer dataLayer = new DataLayer();

public List<Posts> GetListOfSpecificBL(string param)

{

try

{

DataTable table = new DataTable();

List<Posts> listPosts = new List<Posts>();

table = dataLayer.GetListOfSpecific(param);

if (table != null && table.Rows.Count > 0)

{

foreach (DataRow dataRow in table.Rows)

{

Posts post = new Posts();

post.Name = dataRow["Name"].ToString();

post.Email = dataRow["Email"].ToString();

post.Category = dataRow["Category"].ToString();

post.Title = dataRow["Title"].ToString();

post.Content = dataRow["Content"].ToString();

listPosts.Add(post);

}

}

return listPosts;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in GetListOfStudents due to "

+ exception.Message, exception.InnerException);

}

}

public List<Posts> GetListOfPostsBL()

{

try

{

DataTable table = new DataTable();

List<Posts> listPosts = new List<Posts>();

table = dataLayer.GetListOfPosts();

if (table != null && table.Rows.Count > 0)

{

foreach (DataRow dataRow in table.Rows)

{

Posts post = new Posts();

post.Name = dataRow["Name"].ToString();

post.Email = dataRow["Email"].ToString();

post.Category = dataRow["Category"].ToString();

post.Title = dataRow["Title"].ToString();

post.Content = dataRow["Content"].ToString();

listPosts.Add(post);

}

}

return listPosts;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in GetListOfStudents due to "

+ exception.Message, exception.InnerException);

}

}

public string InsertNewUser(Users user)

{

try

{

string response = dataLayer.InsertUserInDB(user);

return response;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in InsertStudent due to "

+ exception.Message, exception.InnerException);

}

}

public string InsertPost(Posts post)

{

try

{

string response = dataLayer.InsertPostInDB(post);

return response;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in InsertStudent due to "

+ exception.Message, exception.InnerException);

}

}

}

}

# Controller

using Microsoft.Ajax.Utilities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Runtime.Remoting.Messaging;

using System.Web.Http;

using System.Web.ModelBinding;

using Web\_API.BusinessLayer;

using Web\_API.Models;

namespace students.Controllers

{

public class StudentsController : ApiController

{

public BL dataLayer = new BL();

[HttpGet]

[Route("GetListOfPosts")]

public List<Posts> GetListOfBootCampPosts()

{

try

{

List<Posts> post = new List<Posts>();

post = dataLayer.GetListOfPostsBL();

return post;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in GetListOfStudents due to "

+ exception.Message, exception.InnerException);

}

}

[HttpGet]

[Route("GetListOfSpecificPosts")]

public List<Posts> GetListOfSpecificBootCampPosts(string param)

{

try

{

List<Posts> post = new List<Posts>();

post = dataLayer.GetListOfSpecificBL(param);

return post;

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in GetListOfStudents due to "

+ exception.Message, exception.InnerException);

}

}

[HttpPost]

[Route("RegisterNewUser")]

public string InsertUser([FromBody]Users user)

{

try

{

if (ModelState.IsValid)

{

string resposne = dataLayer.InsertNewUser(user);

return resposne;

}

else

{

return "Model Is Not Valid";

}

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in InsertStudent due to "

+ exception.Message, exception.InnerException);

}

}

[HttpPost]

[Route("InsertPost")]

public string InsertPost([FromBody] Posts post)

{

try

{

if (ModelState.IsValid)

{

string resposne = dataLayer.InsertPost(post);

return resposne;

}

else

{

return "Model Is Not Valid";

}

}

catch (Exception exception)

{

throw new Exception("An exception of type " + exception.GetType().ToString()

+ " is encountered in InsertStudent due to "

+ exception.Message, exception.InnerException);

}

}

}

}

# Models

using System.ComponentModel.DataAnnotations;

namespace Web\_API.Models

{

public class Users

{

public string Name { get; set; }

public string Email { get; set; }

}

}

using System.ComponentModel.DataAnnotations;

namespace Web\_API.Models

{

public class Emails

{

public string Email { get; set; }

}

}

using System.ComponentModel.DataAnnotations;

namespace Web\_API.Models

{

public class Posts

{

public string Name { get; set; }

public string Email { get; set; }

public string Category { get; set; }

public string Title { get; set; }

public string Content { get; set; }

}

}

Js

$(document).ready(() => {

    var students;

    const POSTS\_API = "http://localhost:62517/GetListOfPosts";

    // fetch and display posts for the first time

    fetchstudents();

    // updateStats();

    // function to render students available in the array students[]

    function displaystudents() {

        $(".posts-container").empty();

        for (let i = 0; i < post.length; i++) {

            $(".posts-container").append(makePost(post[i]));

        }

    }

    function displaySpecificPost() {

        $(".specific-posts-container").empty();

        for (let i = 0; i < post.length; i++) {

            $(".specific-posts-container").append(makePost(post[i]));

        }

    }

    // function to make a post

    function makePost(post) {

        return `

                <div class="post">

                    <p><span style="font-weight:bold;"> User Name:      </span>     ${post.Name}</p></br>

                    <p><span style="font-weight:bold;"> Email:      </span>     ${post.Email}</p></br>

                    <p><span style="font-weight:bold;"> Category:       </span>     ${post.Category}</p></br>

                    <p><span style="font-weight:bold;"> Title:      </span>     ${post.Title}</p></br>

                    <p><span style="font-weight:bold;"> Content:        </span>     ${post.Content}</p>

                </div>

                `;

    }

    function displayError(error) {

        $(".posts-container").html(

            '<div class="error">Some Error has occured!</div>'

        );

    }

    function fetchstudents() {

        $.ajax({

            method: "GET",

            url: POSTS\_API,

            success: (data) => {

                post = data;

                displaystudents();

            },

            error: displayError,

        });

    }

    //Specific User Fetch function

    /////////////////////////////

    // Function to handle form submission

    $("#postForm").submit(function (event) {

        event.preventDefault(); // Prevent the default form submission

        // Gather form data

        const formData = {

            name: $("#name").val(),

            email: $("#email").val(),

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

            title: $("#title").val(),

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

        };

        // Make an AJAX POST request

        $.ajax({

            type: "POST",

            url: "http://localhost:62517/InsertPost", // Replace this with your actual API endpoint URL

            data: JSON.stringify(formData),

            contentType: "application/json",

            success: function (response) {

                // Handle successful response from the server

                alert(response);

                // You can perform additional actions here, like clearing the form or showing a success message.

            },

            error: function (error) {

                // Handle error response from the server

                alert("Failed to add student because of: " + error + ". Please try again!");

                // You can perform additional error handling here.

            },

        });

    });

    $("#userForm").submit(function (event) {

        event.preventDefault(); // Prevent the default form submission

        // Gather form data

        var parameter = $('#uemail').val().trim();

        // Make an AJAX POST request

        $.ajax({

            type: "GET",

            url: "http://localhost:62517/GetListOfSpecificPosts", // Replace this with your actual API endpoint URL

            data: { param: parameter }, // Send the parameter to the server

            dataType: 'json',

                success: (data) => {

                    post = data;

                    displaySpecificPost();

                    if (post.length > 0) {

                        alert("Displaying posts form user '" + parameter+"'");

                    }

                    else {

                        alert("No available post from user '" + parameter+"' or the user doesn't exists!")

                    }

                    // Handle successful response from the server

                // You can perform additional actions here, like clearing the form or showing a success message.

            },

            error: function (error) {

                // Handle error response from the server

                alert("Failed to add student because of: " + error + ". Please try again!");

                // You can perform additional error handling here.

            },

        });

    });

    $("#newUserForm").submit(function (event) {

        event.preventDefault();

        const formData = {

            name: $("#newUname").val(),

            email: $("#newUemail").val(),

        };

        $.ajax({

            type: 'POST',

            url: "http://localhost:62517/RegisterNewUser",

            data: JSON.stringify(formData),

            contentType: "application/json",

            success: function (response) {

                // Handle successful response from the server

                alert(response)

                // You can perform additional actions here, like clearing the form or showing a success message.

            },

            error: function (error) {

                // Handle error response from the server

                alert("Failed to add User because of: " + error + ". Please try again!");

                // You can perform additional error handling here.

            },

        })

    })

});

# Index

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <link rel="stylesheet" href="./src/style.css" />

    <title>BLOG POST</title>

    <style>

        h1 {

            text-align: center;

            margin-top: 20px;

            margin-bottom: 20px;

        }

        form {

            max-width: 400px;

            margin: 0 auto;

            padding: 20px;

            border: 1px solid #ccc;

            border-radius: 5px;

            background-color: #f9f9f9;

        }

        label,

        input {

            display: block;

            margin-bottom: 10px;

            width:max-content;

        }

            input[type="submit"] {

                background-color: #4CAF50;

                color: #fff;

                padding: 10px 20px;

                border: none;

                border-radius: 4px;

                cursor: pointer;

            }

                input[type="submit"]:hover {

                    background-color: #45a049;

                }

        .form-group {

            margin-bottom: 20px;

        }

        .container {

            display: flex;

            justify-content: space-between;

            align-items: flex-start;

        }

    </style>

</head>

<body>

    <!-- navbar -->

    <nav>

        <ul>

            <li>

                <h2><a href="index.html" style="text-decoration: none;">Blog Post</a></h2>

            </li>

            <li>

                <a href="UserRegistration.html" style="text-decoration: none;">User Registration</a>

            </li>

            <li>

                <a href="SpecificUser.html" style="text-decoration: none;">Specific Posts</a>

            </li>

        </ul>

    </nav>

    <!-- posts container -->

    <h1>Create a new Post</h1>

    <form id="postForm">

        <div class="form-group">

            <label for="name">Name:</label>

            <input type="text" name="Name" id="name" required>

        </div>

        <div class="form-group">

            <label for="email">Email:</label>

            <input type="text" name="email" id="email" required>

        </div>

        <div class="form-group">

            <label for="category">Category: (e.g., Science, Arts, Philosphy, History)</label>

            <input type="text" name="category" id="category" required>

        </div>

        <div class="form-group">

            <label for="title">Title:</label>

            <input type="text" name="title" id="title" required>

        </div>

        <div class="form-group">

            <label for="content">Content:</label>

            <textarea style="min-width:350px; min-height: 150px" type="text" name="content" id="content" required></textarea>

        </div>

        <div class="form-group">

            <input type="submit" value="Add Post">

        </div>

    </form>

    <h1>List of Availabe Posts (Latest Posts at Last)</h1>

    <span class="posts-container"></span>

    <!-- scripts -->

    <script src="https://code.jquery.com/jquery-3.7.0.min.js"

            integrity="sha256-2Pmvv0kuTBOenSvLm6bvfBSSHrUJ+3A7x6P5Ebd07/g="

            crossorigin="anonymous"></script>

    <script src="./src/app.js"></script>

</body>

</html>

# Specific user

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <link rel="stylesheet" href="./src/style.css" />

    <title>BLOG POST</title>

    <style>

        h1 {

            text-align: center;

            margin-top: 20px;

            margin-bottom: 20px;

        }

        form {

            max-width: 400px;

            margin: 0 auto;

            padding: 20px;

            border: 1px solid #ccc;

            border-radius: 5px;

            background-color: #f9f9f9;

        }

        label,

        input {

            display: block;

            margin-bottom: 10px;

        }

            input[type="submit"] {

                background-color: #4CAF50;

                color: #fff;

                padding: 10px 20px;

                border: none;

                border-radius: 4px;

                cursor: pointer;

            }

                input[type="submit"]:hover {

                    background-color: #45a049;

                }

        .form-group {

            margin-bottom: 20px;

            margin-top: 20px;

        }

        .container {

            display: flex;

            justify-content: space-between;

            align-items: flex-start;

        }

        .post {

            background-color: white;

            padding: 1rem;

            border-radius: 0.25rem;

            transition-duration: 300ms;

            margin-left:5%;

            margin-right:5%;

            margin-top:2%;

        }

    </style>

</head>

<body>

    <!-- navbar -->

    <nav>

        <ul>

            <li>

                <h2><a href="index.html" style="text-decoration: none;">Blog Post</a></h2>

            </li>

            <li>

                <a href="UserRegistration.html" style="text-decoration: none;">User Registration</a>

            </li>

            <li>

                <a href="SpecificUser.html" style="text-decoration: none;">Specific Posts</a>

            </li>

        </ul>

    </nav>

    <h1>Posts From Specific User</h1>

    <form id="userForm" style="display: flex; justify-content: space-between; align-items: center; box-shadow: 0 0 10px black; ">

        <div class="form-group">

            <label for="uemail">User EMail:</label>

            <input type="text" name="uemail" id="uemail" required>

        </div>

        <div class="form-group">

            <input type="submit" value="Find Post">

        </div>

    </form>

    <!-- posts container -->

    <span class="specific-posts-container"></span>

    <!-- scripts -->

    <script src="https://code.jquery.com/jquery-3.7.0.min.js"

            integrity="sha256-2Pmvv0kuTBOenSvLm6bvfBSSHrUJ+3A7x6P5Ebd07/g="

            crossorigin="anonymous"></script>

    <script src="./src/app.js"></script>

</body>

</html>

# User registeration

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <link rel="stylesheet" href="./src/style.css" />

    <title>BLOG POST</title>

    <style>

        h1 {

            text-align: center;

            margin-top: 20px;

            margin-bottom: 20px;

        }

        form {

            max-width: 400px;

            margin: 0 auto;

            padding: 20px;

            border: 1px solid #ccc;

            border-radius: 5px;

            background-color: #f9f9f9;

        }

        label,

        input {

            display: block;

            margin-bottom: 10px;

        }

            input[type="submit"] {

                background-color: #4CAF50;

                color: #fff;

                padding: 10px 20px;

                border: none;

                border-radius: 4px;

                cursor: pointer;

            }

                input[type="submit"]:hover {

                    background-color: #45a049;

                }

        .form-group {

            margin-bottom: 20px;

            margin-top: 20px;

        }

        .container {

            display: flex;

            justify-content: space-between;

            align-items: flex-start;

        }

        .post {

            background-color: white;

            padding: 1rem;

            border-radius: 0.25rem;

            transition-duration: 300ms;

            margin-left:5%;

            margin-right:5%;

            margin-top:2%;

        }

    </style>

</head>

<body>

    <!-- navbar -->

    <nav>

        <ul>

            <li>

                <h2><a href="index.html" style="text-decoration: none;">Blog Post</a></h2>

            </li>

            <li>

                <a href="UserRegistration.html" style="text-decoration: none;">User Registration</a>

            </li>

            <li>

                <a href="SpecificUser.html" style="text-decoration: none;">Specific Posts</a>

            </li>

        </ul>

    </nav>

    <h1>New User Registration</h1>

    <form id="newUserForm" style="box-shadow: 0 0 10px black; ">

        <div class="form-group">

            <label for="newUemail">Your Email:</label>

            <input type="text" name="newUemail" id="newUemail" required>

        </div>

        <div class="form-group">

            <label for="newUname">Your Name:</label>

            <input type="text" name="newUname" id="newUname" required>

        </div>

        <div class="form-group">

            <input type="submit" value="Register">

        </div>

    </form>

    <!-- scripts -->

    <script src="https://code.jquery.com/jquery-3.7.0.min.js"

            integrity="sha256-2Pmvv0kuTBOenSvLm6bvfBSSHrUJ+3A7x6P5Ebd07/g="

            crossorigin="anonymous"></script>

    <script src="./src/app.js"></script>

</body>

</html>