

SIA102 FINAL EXAM GUIDE

Create table of “Department and Student” using Create Table query.

```
Create table Department(  
    DepartmentID int not null Identity(1,1),  
    DeptName varchar(255) not null,  
    DeptDescription varchar(255) not null,  
    DepartmentDean varchar(255)  
);  
  
CREATE TABLE Student(  
    StudentID int IDENTITY(1,1) NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    DepartmentID int NULL  
);
```

Insert an instance into Student table.

Insert at least 10 students.

```
INSERT INTO Student (LastName, FirstName)  
VALUES  
( 'James', 'LeBron' ),  
( 'Curry', 'Stephen' ),  
( 'Durant', 'Kevin' ),  
( 'Antetokounmpo', 'Giannis' ),  
( 'Jokic', 'Nikola' ),  
( 'Doncic', 'Luka' ),  
( 'Tatum', 'Jayson' ),  
( 'Booker', 'Devin' ),  
( 'Butler', 'Jimmy' ),  
( 'Morant', 'Ja' );
```

On the project, add a model for department and named it “Department” and paste the code given.

```
namespace WebApplication1.Models  
{  
  
    public class Department  
    {  
        public int DepartmentID { get; set; }  
  
        public string DeptName { get; set; } = "";  
  
        public string DeptDescription { get; set; } = "";  
  
        public string? DeanFirstName { get; set; }  
  
        public string? DeanLastName { get; set; }  
  
        public string? DepartmentDean { get; set; }  
    }  
}
```

```

    }
}

```

Add another model for student. Named it "StudentInfo" and paste the code given.

```

namespace WebApplication1.Models
{
    public class StudentInfo
    {
        public string StudentID { get; set; }
        public string LastName { get; set; }
        public string FirstName { get; set; }
        public string DeptName { get; set; }
    }
}

```

Add page for department and named is as "Department" and paste the given code.

```

@page
@model WebApplication1.Pages.DepartmentModel

<h2>Departments</h2>

<button class="btn btn-success" data-toggle="modal" data-
target="#addDepartmentModal">Add New Department</button>

<table class="table">
    <thead>
        <tr>
            <th>Department</th>
            <th>Department Description</th>
            <th>Dean</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var department in Model.Departments)
        {
            <tr>
                <td>@department.DeptName</td>
                <td>@department.DeptDescription</td>
                <td>@department.DeanFirstName @department.DeanLastName</td>
                <td>
                    <button class="btn btn-primary edit-button"
                        data-toggle="modal"
                        data-target="#editDepartmentModal"
                        data-id="@department.DepartmentID"
                        data-name="@department.DeptName"
                        data-description="@department.DeptDescription"
                        data-deanid="@department.DepartmentDean">
                        Edit
                    </button>
                </td>
            </tr>
        }
    </tbody>
</table>

```

```

        </td>
    </tr>
}
</tbody>
</table>
<div class="modal fade" id="addDepartmentModal" tabindex="-1" role="dialog" aria-
labelledby="addDepartmentModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="addDepartmentModalLabel">Add New
Department</h5>
                <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form method="post">
                <div class="modal-body">
                    <div class="form-group">
                        <label for="DeptName">Department Name</label>
                        <input type="text" class="form-control" id="DeptName"
name="DeptName" required />
                    </div>
                    <div class="form-group">
                        <label for="DeptDescription">Department Description</label>
                        <input type="text" class="form-control" id="DeptDescription"
name="DeptDescription" required />
                    </div>
                    <div class="form-group">
                        <label for="DepartmentDean">Department Dean</label>
                        <select class="form-control" id="DepartmentDean"
name="DepartmentDean" required>
                            <option value="">Select a Dean</option>
                            @foreach (var user in Model.Users)
                            {
                                <option value="@user.Id">@user.FirstName
@user.LastName</option>
                            }
                        </select>
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
                    <button type="submit" class="btn btn-primary">Add
Department</button>
                </div>
            </form>
        </div>
    </div>
</div>
<div class="modal fade" id="editDepartmentModal" tabindex="-1" role="dialog" aria-
labelledby="editDepartmentModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">

```

```

        <h5 class="modal-title" id="editDepartmentModalLabel">Edit
Department</h5>
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>
    <form method="post">
        <input type="hidden" id="EditDepartmentID" name="DepartmentID" />
        <div class="modal-body">
            <div class="form-group">
                <label for="EditDeptName">Department Name</label>
                <input type="text" class="form-control" id="EditDeptName"
name="DeptName" required />
            </div>
            <div class="form-group">
                <label for="EditDeptDescription">Department
Description</label>
                <input type="text" class="form-control"
id="EditDeptDescription" name="DeptDescription" required />
            </div>
            <div class="form-group">
                <label for="EditDepartmentDean">Department Dean</label>
                <select class="form-control" id="EditDepartmentDean"
name="DepartmentDean" required>
                    <option value="">Select a Dean</option>
                    @foreach (var user in Model.Users)
                    {
                        <!--option value="@user.Id" @(user.Id ==
Model.DepartmentDeanId ? "selected" : "")>
                            @user.FirstName @user.LastName
                        </!option>
                    }
                </select>
            </div>
        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
            <button type="submit" class="btn btn-primary">Update
Department</button>
        </div>
    </form>
</div>
</div>
</div>
<script>
    document.addEventListener('DOMContentLoaded', function () {
        const editButtons = document.querySelectorAll('.edit-button');
        editButtons.forEach(button => {
            button.addEventListener('click', function () {
                const departmentId = this.getAttribute('data-id');
                const deptName = this.getAttribute('data-name');
                const deptDescription = this.getAttribute('data-description');
                const departmentDeanId = this.getAttribute('data-deanid');

                // Populate the edit modal fields

```

```

        document.getElementById('EditDepartmentID').value = departmentId;
        document.getElementById('EditDeptName').value = deptName;
        document.getElementById('EditDeptDescription').value =
deptDescription;

        const editDeanSelect =
document.getElementById('EditDepartmentDean');
        Array.from(editDeanSelect.options).forEach(option => {
            option.selected = option.value === departmentDeanId;
        });
    });
});
</script>

```

Paste this code on "Department.cshtml.cs".

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
using Microsoft.Data.SqlClient;
using WebApplication1.Models;
using Microsoft.Extensions.Configuration;
using System.Collections.Generic;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;

namespace WebApplication1.Pages
{
    [Authorize(Roles = "admin")]
    public class DepartmentModel : PageModel
    {
        private readonly string _connectionString;

        public List<Department> Departments { get; set; } = new List<Department>();
        public List<ApplicationUser> Users { get; set; } = new
List<ApplicationUser>();
        public string DepartmentDeanId { get; set; }

        [BindProperty]
        public string DeptName { get; set; }
        [BindProperty]
        public string DeptDescription { get; set; }
        [BindProperty]
        public string DepartmentDean { get; set; }
        [BindProperty]
        public int DepartmentID { get; set; }

        public DepartmentModel(IConfiguration configuration)
        {
            _connectionString =
configuration.GetConnectionString("DefaultConnection")
            ?? throw new ArgumentNullException("Connection string
'DefaultConnection' not found.");
        }

        public async Task OnGetAsync()
        {

```

```

        await LoadDepartmentsAsync();
        await LoadUsersAsync();
    }

    private async Task LoadDepartmentsAsync()
    {
        using var connection = new SqlConnection(_connectionString);
        await connection.OpenAsync();
        var query = @"SELECT
                        department.DepartmentID,
                        department.DeptName,
                        department.DeptDescription,
                        AspNetUsers.FirstName,
                        AspNetUsers.LastName,
                        department.DepartmentDean
                    FROM
                        department
                    LEFT JOIN
                        AspNetUsers ON department.DepartmentDean =
AspNetUsers.Id";

        using var command = new SqlCommand(query, connection);
        using var reader = await command.ExecuteReaderAsync();

        Departments.Clear();
        while (await reader.ReadAsync())
        {
            Departments.Add(new Department
            {
                DepartmentID = reader.GetInt32(0),
                DeptName = reader.GetString(1),
                DeptDescription = reader.GetString(2),
                DeanFirstName = reader.IsDBNull(3) ? "Null" :
reader.GetString(3),
                DeanLastName = reader.IsDBNull(4) ? "Null" :
reader.GetString(4),
                DepartmentDean = reader.IsDBNull(5) ? null : reader.GetString(5)
            });
        }
    }

    private async Task LoadUsersAsync()
    {
        using var connection = new SqlConnection(_connectionString);
        await connection.OpenAsync();
        var query = "SELECT Id, FirstName, LastName FROM AspNetUsers";

        using var command = new SqlCommand(query, connection);
        using var reader = await command.ExecuteReaderAsync();

        Users.Clear();
        while (await reader.ReadAsync())
        {
            Users.Add(new ApplicationUser
            {
                Id = reader.GetString(0),
                FirstName = reader.GetString(1),
                LastName = reader.GetString(2)
            });
        }
    }

```

```

        });
    }
}

public async Task<IActionResult> OnPostAsync()
{
    if (DepartmentID == 0)
    {
        await AddDepartmentAsync();
    }
    else
    {
        await EditDepartmentAsync();
    }

    return RedirectToPage();
}

private async Task AddDepartmentAsync()
{
    using var connection = new SqlConnection(_connectionString);
    await connection.OpenAsync();

    var query = @"INSERT INTO department (DeptName, DeptDescription,
DepartmentDean)
                VALUES (@DeptName, @DeptDescription, @DepartmentDean)";
    using var command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@DeptName", DeptName);
    command.Parameters.AddWithValue("@DeptDescription", DeptDescription);
    command.Parameters.AddWithValue("@DepartmentDean", DepartmentDean);
    await command.ExecuteNonQueryAsync();
}

private async Task EditDepartmentAsync()
{
    using var connection = new SqlConnection(_connectionString);
    await connection.OpenAsync();

    var query = @"UPDATE department
                SET DeptName = @DeptName, DeptDescription =
@DeptDescription, DepartmentDean = @DepartmentDean
                WHERE DepartmentID = @DepartmentID";
    using var command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@DeptName", DeptName);
    command.Parameters.AddWithValue("@DeptDescription", DeptDescription);
    command.Parameters.AddWithValue("@DepartmentDean", DepartmentDean);
    command.Parameters.AddWithValue("@DepartmentID", DepartmentID);

    await command.ExecuteNonQueryAsync();
}
}
}

```

Add student page. Named it “Student” and paste the code.

```
@page
@model WebApplication1.Pages.StudentModel
@{
    ViewData["Title"] = "Students";
}

@using Microsoft.AspNetCore.Identity
@using WebApplication1.Models

@inject SignInManager<ApplicationUser> SignInManager
@inject UserManager<ApplicationUser> UserManager

<h2>Students</h2>

@if (SignInManager.IsSignedIn(User))
{
    if (User.IsInRole("admin"))
    {
        <table class="table">
            <thead>
                <tr>
                    <th>Last Name</th>
                    <th>First Name</th>
                    <th>Department Name</th>
                    <th>Action</th>
                </tr>
            </thead>
            <tbody>
                @foreach (var student in Model.Students)
                {
                    <tr>
                        <td>@student.LastName</td>
                        <td>@student.FirstName</td>
                        <td>@student.DeptName</td>
                        <td>
                            <button class="btn btn-primary" data-bs-toggle="modal"
data-bs-target="#editModal"
                                onclick="populateModal('@student.StudentID',
                                '@student.LastName', '@student.FirstName', '@student.DeptName')">
                                Edit
                            </button>
                        </td>
                    </tr>
                }
            </tbody>
        </table>
    }
    else if (User.IsInRole("client"))
    {
        <table class="table">
            <thead>
                <tr>
                    <th>Last Name</th>
```



```

        <th>First Name</th>
        <th>Department Name</th>
    </tr>
</thead>
<tbody>
    @foreach (var student in Model.Students)
    {
        <tr>
            <td>@student.LastName</td>
            <td>@student.FirstName</td>
            <td>@student.DeptName</td>
        </tr>
    }
</tbody>
</table>
    }
}

```

```

<div class="modal fade" id="editModal" tabindex="-1" aria-
labelledby="editModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <form method="post">
                <div class="modal-header">
                    <h5 class="modal-title" id="editModalLabel">Edit Student</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    <input type="hidden" id="StudentID" name="StudentID" />
                    <div class="mb-3">
                        <label for="LastName" class="form-label">Last Name</label>
                        <input type="text" class="form-control" id="LastName"
name="LastName" />
                    </div>
                    <div class="mb-3">
                        <label for="FirstName" class="form-label">First Name</label>
                        <input type="text" class="form-control" id="FirstName"
name="FirstName" />
                    </div>
                    <div class="mb-3">
                        <label for="DepartmentID" class="form-
label">Department</label>
                        <select class="form-select" id="DepartmentID"
name="DepartmentID">
                            @foreach (var department in Model.Departments)
                            {
                                <option
value="@department.DepartmentID">@department.DeptName</option>
                            }
                        </select>
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>

```



```

        await LoadStudentsAsync();
        await LoadDepartmentsAsync();
    }

    public async Task<IActionResult> OnPostAsync(string StudentID, string
LastName, string FirstName, int DepartmentID)
    {
        if (string.IsNullOrEmpty(StudentID) || string.IsNullOrEmpty(LastName) ||
string.IsNullOrEmpty(FirstName) || DepartmentID <= 0)
        {
            ModelState.AddModelError(string.Empty, "Invalid input.");
            await LoadStudentsAsync();
            await LoadDepartmentsAsync();
            return Page();
        }

        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            await connection.OpenAsync();

            string query = @"
                UPDATE Student
                SET
                    LastName = @LastName,
                    FirstName = @FirstName,
                    DepartmentID = @DepartmentID
                WHERE StudentID = @StudentID";

            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@StudentID", StudentID);
                command.Parameters.AddWithValue("@LastName", LastName);
                command.Parameters.AddWithValue("@FirstName", FirstName);
                command.Parameters.AddWithValue("@DepartmentID", DepartmentID);

                await command.ExecuteNonQueryAsync();
            }
        }

        return RedirectToPage();
    }

    private async Task LoadStudentsAsync()
    {
        string userId = User.FindFirst(ClaimTypes.NameIdentifier)?.Value;
        bool isAdmin = User.FindFirst(ClaimTypes.Role)?.Value == "admin";
        using (SqlConnection connection = new SqlConnection(_connectionString))
        {
            await connection.OpenAsync();

            string query;

            if (isAdmin)
            {
                query = @"
                    SELECT
                        Student.StudentID,

```

```

        Student.LastName,
        Student.FirstName,
        Department.DeptName
    FROM
        Student
    LEFT JOIN
        Department ON Student.DepartmentID =
Department.DepartmentID
    Order By
        Student.LastName";
    }
    else
    {
        query = @"
        SELECT
            Student.StudentID,
            Student.LastName,
            Student.FirstName,
            Department.DeptName
        FROM
            Student
        LEFT JOIN
            Department ON Student.DepartmentID =
Department.DepartmentID
        LEFT JOIN
            AspNetUsers ON Department.DepartmentDean =
AspNetUsers.Id
        WHERE
            AspNetUsers.Id = @UserId
        Order By
            Student.LastName";
    }

    using (SqlCommand command = new SqlCommand(query, connection))
    {
        if (!isAdmin && !string.IsNullOrEmpty(userId))
        {
            command.Parameters.Add(new SqlParameter("@UserId", userId));
        }

        using (SqlDataReader reader = await
command.ExecuteReaderAsync())
        {
            while (await reader.ReadAsync())
            {
                Students.Add(new StudentInfo
                {
                    StudentID = reader["StudentID"].ToString(),
                    LastName = reader["LastName"].ToString(),
                    FirstName = reader["FirstName"].ToString(),
                    DeptName = reader["DeptName"].ToString()
                });
            }
        }
    }
}
}
}
}

```

```

private async Task LoadDepartmentsAsync()
{
    using (SqlConnection connection = new SqlConnection(_connectionString))
    {
        await connection.OpenAsync();

        string query = "SELECT DepartmentID, DeptName FROM Department";

        using (SqlCommand command = new SqlCommand(query, connection))
        {
            using (SqlDataReader reader = await
command.ExecuteReaderAsync())
            {
                while (await reader.ReadAsync())
                {
                    Departments.Add(new Department
                    {
                        DepartmentID = (int)reader["DepartmentID"],
                        DeptName = reader["DeptName"].ToString()
                    });
                }
            }
        }
    }
}

```

Now add Department and Student option on the admin side.

Pages – Shared - _Layout.cshtml.

Congratulations!!!