Academic: Create

<nav class="subnavigate">

<ul>

<li>

<**a** **asp-action**="AcademicYears">Manage Academic Year</**a**>

</li>

<li class="active">

<**a** **asp-action**="AcademicYearCreate">Create New</**a**>

</li>

@if (TempData["ErrorMessage"] != null)

{

<div class="alert alert-danger" role="alert">

@TempData["ErrorMessage"]

</div>

}

</ul>

</nav>

<div id="create\_new">

<div class="main\_information">

<h3>Create New Academic Year</h3>

<**form** **asp-action**="AcademicYearCreate" method="post" onsubmit="return validateForm()">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<div class="author\_article ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

<label for="academicYearInput">Academic Year:</label>

<div class="author\_name bg-white p-3 mb-2 rounded">

<**input** **asp-for**="AcademicYears"

id="academicYearInput"

**type**="text"

class="form-control border border-white"

aria-describedby="academicYearInputHelp"

placeholder="Input Academic Name (5-50 characters with no special characters)"

required

maxlength="50"

minlength="5"

pattern="[A-Za-z0-9\s]\*"

oninvalid="this.setCustomValidity('Please enter the correct requested information')"

oninput="this.setCustomValidity('')">

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

<span class="text-danger">@Html.ValidationMessageFor(model => model.AcademicYears)</span>

</div>

</p>

</div>

<div class="last\_modified ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Start Date: <br>

<div class="date\_time\_modified bg-white p-3 mb-2 rounded">

<**input** **asp-for**="StartDate"

class="form-control border border-white"

placeholder="MM/DD/YY"

aria-describedby="addon-wrapping"

required

oninvalid="this.setCustomValidity('Please enter Start Date')"

oninput="this.setCustomValidity('')">

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

</div>

</p>

</div>

<div class="feedback\_content ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Closure Date: <br>

<div class="content\_container bg-white p-3 mb-2 rounded">

<**input** **asp-for**="ClosureDate"

class="form-control text-break border border-white"

placeholder="MM/DD/YY"

aria-describedby="addon-wrapping"

required

oninvalid="this.setCustomValidity('Please enter Closure Date')"

oninput="this.setCustomValidity('')">

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

</div>

</p>

</div>

<**form** id="saveForm" **asp-action**="AcademicYearCreate" method="post">

<!-- Các trường dữ liệu khác -->

<div class="btn\_group ps-4 mt-5">

<button type="submit" class="btn btn-outline-success btn-lg me-3">

<i class="fa-regular fa-floppy-disk"></i> Save

</button>

<**a** **asp-action**="AcademicYears" type="button" class="btn btn-outline-secondary btn-lg">

<i class="fa-solid fa-arrow-left"></i> Back

</**a**>

</div>

</**form**>

</**form**>

</div>

</div>

Script:

<script>

document.addEventListener("DOMContentLoaded", function () {

const filterMobileBtn = document.querySelector(".sidebar\_mobile");

const hideFilterBtn = document.querySelector(".hideSidebar");

const menuSidebar = document.querySelector(".menu\_sidebar");

if (filterMobileBtn && hideFilterBtn && menuSidebar) {

filterMobileBtn.addEventListener("click", function (event) {

event.preventDefault();

showSidebar();

});

hideFilterBtn.addEventListener("click", function (event) {

event.preventDefault();

hideSidebar();

});

}

// Ẩn menu\_sidebar khi trang được tải

if (menuSidebar) {

menuSidebar.style.display = "none";

}

// Gán sự kiện submit form

document.querySelector("form").addEventListener("submit", function (event) {

event.preventDefault(); // Ngăn chặn hành động mặc định của form

// Thực hiện gửi dữ liệu form bằng AJAX

var formData = new FormData(this);

var xhr = new XMLHttpRequest();

xhr.open("POST", this.action);

xhr.onload = function () {

if (xhr.status === 200) {

// Kiểm tra nếu dữ liệu trả về là thành công

var response = JSON.parse(xhr.responseText);

if (response.success) {

// Hiển thị modal thành công

var successModal = new bootstrap.Modal(document.getElementById('successModal'));

successModal.show();

} else {

// Xử lý logic khi lưu không thành công (nếu cần)

}

} else {

// Xử lý logic khi có lỗi trong quá trình gửi dữ liệu (nếu cần)

}

};

xhr.send(formData);

});

});

function redirectToIndex() {

window.location.href = "/Admin/AcademicYears"; // Thay đổi đường dẫn tương ứng với trang index của bạn

}

function showSidebar() {

const menuSidebar = document.querySelector(".menu\_sidebar");

if (menuSidebar) {

menuSidebar.style.display = "block";

const hideFilterBtn = document.querySelector(".hideSidebar");

if (hideFilterBtn) {

hideFilterBtn.classList.add("show");

}

}

}

function hideSidebar() {

const menuSidebar = document.querySelector(".menu\_sidebar");

if (menuSidebar) {

menuSidebar.style.display = "none";

}

}

</script>

Controller : CreateAca

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> AcademicYearCreate([Bind("AcademicYearID,AcademicYears,StartDate,ClosureDate")] AcademicYear academicYear)

{

if (ModelState.IsValid)

{

// Kiểm tra xem thông tin nhập vào có trùng lặp không

var existingAcademicYear = await \_context.AcademicYears.FirstOrDefaultAsync(a => a.AcademicYears == academicYear.AcademicYears);

if (existingAcademicYear != null)

{

// Nếu có trùng lặp, hiển thị thông báo lỗi

ModelState.AddModelError("AcademicYears", "This academic year already exists.");

return View("~/Views/Admin/AcademicYears/Create.cshtml", academicYear);

}

// Nếu không có trùng lặp, thêm mới vào cơ sở dữ liệu

\_context.Add(academicYear);

await \_context.SaveChangesAsync();

// Đặt thông báo thành công vào TempData

TempData["SuccessMessage"] = "An Academic Year has been created successfully!";

// Chuyển hướng người dùng về trang Index

return RedirectToAction(nameof(AcademicYears));

}

return View("~/Views/Admin/AcademicYears/Create.cshtml", academicYear);

}

Controler: editAca

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> AcademicYearEdit(int id, [Bind("AcademicYearID,AcademicYears,StartDate,ClosureDate")] AcademicYear academicYear)

{

if (id != academicYear.AcademicYearID)

{

return NotFound();

}

if (ModelState.IsValid)

{

try

{

\_context.Update(academicYear);

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!AcademicYearExists(academicYear.AcademicYearID))

{

return NotFound();

}

else

{

throw;

}

}

TempData["UpdateSuccessMessage"] = "Update successfully!";

return RedirectToAction(nameof(AcademicYears));

}

return View(academicYear);

}

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

Controller : Delete Confirm

[HttpPost, ActionName("AcademicYearDelete")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> AcademicYearDeleteConfirmed(int id)

{

if (\_context.AcademicYears == null)

{

return Problem("Entity set 'ApplicationDbContext.AcademicYears' is null.");

}

var academicYear = await \_context.AcademicYears.FindAsync(id);

if (academicYear != null)

{

\_context.AcademicYears.Remove(academicYear);

await \_context.SaveChangesAsync();

// Thiết lập TempData để lưu thông báo xóa thành công

TempData["DeleteSuccessMessage"] = "Academic year has been deleted successfully.";

}

else

{

// Handle the case if academic year is not found

}

return RedirectToAction(nameof(Index));

}

---------------------------------Semester----------------------------------------

Controller : Create ( 2 cai )

public IActionResult SemesterCreate()

{

ViewData["AcademicYearID"] = new SelectList(\_context.AcademicYears, "AcademicYears", "AcademicYears");

ViewData["FacultyID"] = new SelectList(\_context.Faculties, "FacultyName", "FacultyName");

return View("~/Views/Admin/Semesters/Create.cshtml");

}

// POST: Semesters/Create

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> SemesterCreate([Bind("SemesterID,SemesterName,Notification,CreatedDate,ClosureDate,Dl1,DL2,FacultyName,AcademicYears")] Semester semester)

{

if (ModelState.IsValid)

{

\_context.Add(semester);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

ViewData["AcademicYearID"] = new SelectList(\_context.AcademicYears, "AcademicYears", "AcademicYears", semester.AcademicYear);

ViewData["FacultyID"] = new SelectList(\_context.Faculties, "FacultyName", "FacultyName", semester.Faculty);

return View(semester);

}

---------Controller : Edit seme ( 2 cai)-----------

public async Task<IActionResult> SemesterEdit(int? id)

{

if (id == null || \_context.Semesters == null)

{

return NotFound();

}

var semester = await \_context.Semesters.FindAsync(id);

if (semester == null)

{

return NotFound();

}

ViewData["AcademicYearID"] = new SelectList(\_context.AcademicYears, "AcademicYears", "AcademicYears", semester.AcademicYear);

ViewData["FacultyID"] = new SelectList(\_context.Faculties, "FacultyName", "FacultyName", semester.Faculty);

return View("~/Views/Admin/Semesters/Edit.cshtml", semester);

}

// POST: Semesters/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> SemesterEdit(int id, [Bind("SemesterID,SemesterName,Notification,CreatedDate,ClosureDate,Dl1,DL2,FacultyID,AcademicYearID")] Semester semester)

{

if (id != semester.SemesterID)

{

return NotFound();

}

if (ModelState.IsValid)

{

try

{

\_context.Update(semester);

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!SemesterExists(semester.SemesterID))

{

return NotFound();

}

else

{

throw;

}

}

return RedirectToAction(nameof(Semesters));

}

ViewData["AcademicYearID"] = new SelectList(\_context.AcademicYears, "AcademicYears", "AcademicYears", semester.AcademicYear);

ViewData["FacultyID"] = new SelectList(\_context.Faculties, "FacultyName", "FacultyName", semester.Faculty);

return View(semester);

}

----------Controller : Delete ( 2 cai )---------------

public async Task<IActionResult> SemesterDelete(int? id)

{

if (id == null || \_context.Semesters == null)

{

return NotFound();

}

var semester = await \_context.Semesters

.Include(s => s.AcademicYear)

.Include(s => s.Faculty)

.FirstOrDefaultAsync(m => m.SemesterID == id);

if (semester == null)

{

return NotFound();

}

return View("~/Views/Admin/Semesters/Delete.cshtml", semester);

}

// POST: Semesters/Delete/5

[HttpPost, ActionName("SemesterDelete")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> SemesterDeleteConfirmed(int id)

{

if (\_context.Semesters == null)

{

return Problem("Entity set 'ApplicationDbContext.Semesters' is null.");

}

var semester = await \_context.Semesters.FindAsync(id);

if (semester != null)

{

\_context.Semesters.Remove(semester);

}

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Semesters));

}

---------View.Create-------------

<div id="create\_new">

<**form** **asp-action**="SemesterCreate">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<div class="main\_information d-flex">

<div class="left\_semester flex-fill">

<h3>Create New Semester</h3>

<div class="author\_article ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Semester Name:<br>

<div class="author\_name bg-white p-3 mb-2 rounded">

<**input** **asp-for**="SemesterName" **type**="text"

class="form-control border border-white"

aria-describedby="addon-wrapping"

placeholder="Input Academic Name (5-50 characters with no special characters)"

required

maxlength="50"

minlength="5"

pattern="[A-Za-z0-9\s]\*"

oninvalid="this.setCustomValidity('Please enter the correct requested information')"

oninput="this.setCustomValidity('')">

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

</div>

</p>

</div>

<div class="status\_publish ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Academic Year Name: <br>

<div class="status bg-white p-3 mb-2 rounded">

<**select** **asp-for**="AcademicYearID" class="form-control" **asp-items**="ViewBag.AcademicYearID"></**select**>

</div>

</p>

</div>

<div class="status\_publish ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Faculty Name: <br>

<div class="status bg-white p-3 mb-2 rounded">

<**select** **asp-for**="FacultyID" class="form-control" **asp-items**="ViewBag.FacultyID"></**select**>

</div>

</p>

</div>

<div class="last\_modified ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Start Date: <br>

<div class="date\_time\_modified bg-white p-3 mb-2 rounded">

<**input** **asp-for**="CreatedDate" **type**="date"

class="form-control border border-white"

placeholder="MM/DD/YY"

aria-describedby="addon-wrapping">

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

</div>

</p>

</div>

<div class="feedback\_content ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Closure Date: <br>

<div class="content\_container bg-white p-3 mb-2 rounded">

<**input** **asp-for**="ClosureDate"

class="form-control text-break border border-white"

placeholder="MM/DD/YY"

aria-describedby="addon-wrapping">

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

</div>

</p>

</div>

<div class="btn\_group ps-4 mt-5">

<button type="submit" class="btn btn-outline-success btn-lg me-3">

Save

</button>

<**a** **asp-action**="Semesters"

type="button"

class="btn btn-outline-secondary btn-lg">

<i class="fa-solid fa-arrow-left"></i> Back to Created

</a>

</div>

</div>

<div class="right\_semester flex-fill">

<div class="note\_content ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Note: <br>

<div class="content\_container bg-white p-3 mb-2 rounded">

<textarea asp-for="Notification" name="" id="" cols="61" rows="15"></textarea>

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

</div>

</p>

</div>

<div class="dateline\_1 ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Deadline 1: <br>

<div class="date\_time\_modified bg-white p-3 mb-2 rounded">

<input asp-for="Dl1" type="date"

class="form-control border border-white"

placeholder="MM/DD/YY"

aria-describedby="addon-wrapping">

<span asp-validation-for="Dl1" class="text-danger"></span>

</div>

</p>

</div>

<div class="dateline\_2 ps-4">

<p class="d-block p-2 fs-5 text fw-bold">

Deadline 2: <br>

<div class="date\_time\_modified bg-white p-3 mb-2 rounded">

<input asp-for="DL2"

class="form-control border border-white"

placeholder="MM/DD/YY"

aria-describedby="addon-wrapping">

<span asp-validation-for="DL2" class="text-danger"></span>

</div>

</p>

</div>

</div>

</div>

</**form**>

</div>

----Script------

<script>

document.addEventListener("DOMContentLoaded", function () {

const filterMobileBtn = document.querySelector(".sidebar\_mobile");

const hideFilterBtn = document.querySelector(".hideSidebar");

const menuSidebar = document.querySelector(".menu\_sidebar");

if (filterMobileBtn && hideFilterBtn && menuSidebar) {

filterMobileBtn.addEventListener("click", function (event) {

event.preventDefault();

showSidebar();

});

hideFilterBtn.addEventListener("click", function (event) {

event.preventDefault();

hideSidebar();

});

}

// Ẩn menu\_sidebar khi trang được tải

if (menuSidebar) {

menuSidebar.style.display = "none";

}

});

function showSidebar() {

const menuSidebar = document.querySelector(".menu\_sidebar");

if (menuSidebar) {

menuSidebar.style.display = "block";

const hideFilterBtn = document.querySelector(".hideSidebar");

if (hideFilterBtn) {

hideFilterBtn.classList.add("show");

}

}

}

function hideSidebar() {

const menuSidebar = document.querySelector(".menu\_sidebar");

if (menuSidebar) {

menuSidebar.style.display = "none";

}

}

</script>

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

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