A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

using Microsoft.AspNetCore.Mvc;

using jouw\_projectnaam.Data;

using jouw\_projectnaam.Domain.Entities;

using System.Linq;

using System.Threading.Tasks;

public class StudentsController : Controller

{

private readonly ApplicationDbContext \_context;

public StudentsController(ApplicationDbContext context)

{

\_context = context;

}

// ========== R - Read (Lezen) ==========

// GET: /Students

// Toont een lijst van alle studenten

public IActionResult Index()

{

var students = \_context.Students.ToList();

return View(students);

}

// GET: /Students/Details/1

// Toont de details van een enkele student

public IActionResult Details(int id)

{

var student = \_context.Students.FirstOrDefault(s => s.Id == id);

if (student == null) return NotFound();

return View(student);

}

// ========== C - Create (Aanmaken) ==========

// GET: /Students/Create

// Toont het formulier om een nieuwe student aan te maken

[HttpGet]

public IActionResult Create()

{

return View();

}

// POST: /Students/Create

// Verwerkt het ingevulde formulier en slaat de nieuwe student op

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create(Student student)

{

if (ModelState.IsValid)

{

\_context.Add(student);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

return View(student);

}

// ========== U - Update (Bewerken) ==========

// GET: /Students/Edit/1

// Toont het formulier om een bestaande student te bewerken

[HttpGet]

public IActionResult Edit(int id)

{

var student = \_context.Students.FirstOrDefault(s => s.Id == id);

if (student == null) return NotFound();

return View(student);

}

// POST: /Students/Edit/1

// Verwerkt het bewerkte formulier en slaat de wijzigingen op

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Edit(int id, Student student)

{

if (id != student.Id) return NotFound();

if (ModelState.IsValid)

{

try

{

\_context.Update(student);

await \_context.SaveChangesAsync();

}

catch (System.Exception)

{

if (!\_context.Students.Any(s => s.Id == student.Id))

{

return NotFound();

}

else

{

throw;

}

}

return RedirectToAction(nameof(Index));

}

return View(student);

}

// ========== D - Delete (Verwijderen) ==========

// GET: /Students/Delete/1

// Toont de bevestigingspagina voor het verwijderen

[HttpGet]

public IActionResult Delete(int id)

{

var student = \_context.Students.FirstOrDefault(s => s.Id == id);

if (student == null) return NotFound();

return View(student);

}

// POST: /Students/Delete/1

// Voert de verwijdering uit na bevestiging

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> DeleteConfirmed(int id)

{

var student = \_context.Students.FirstOrDefault(s => s.Id == id);

if (student != null)

{

\_context.Students.Remove(student);

await \_context.SaveChangesAsync();

}

return RedirectToAction(nameof(Index));

}

}