Міністерство освіти України

Національний технічний університет "ХПІ"

кафедра "Інформатики та інтелектуальної власності"

**Звіт**

**Лабораторна робота 7**

з дисципліни "Проєктування серверних застосунків"

Виконав: студент групи КН-922В

Шершньов М.О.

Перевірив:

Івашко А.В.

Харків 2025

Зміст

[Завдання 3](#_Toc195004350)

[Класи controllervalidation, modelnotation 4](#_Toc195004351)

[Клас ControllerValidation 4](#_Toc195004352)

[Клас ModelNotation 12](#_Toc195004353)

[Висновок 16](#_Toc195004354)

**Використання засобів валідації (перевірки на коректність) інформації під час її обробки в MVC WEB-застосунку**

**Мета роботи:** придбати практику у використанні різноманітних засобів валідації інформації під час її додання редагування та аналізу у формах та БД МѴС застосунків

Завдання

1. Модифікувати підсумкові завдання (6 та 9) минулої роботи, додавши до їхніх проектів засоби валідації даних через контролери. Розміщення повідомлень про помилки валідації витримати зверху від відповідних полів форми.

2. Модифікувати підсумкові завдання (6 та 9) минулої роботи, додавши до їхніх проектів засоби валідації даних через нотації моделі. Розміщення повідомлень про помилки валідації витримати зверху від відповідних полів форми.

Класи для роботи:

1. «Валідація у контролері» - **ControllerValidation**
2. «Нотація у моделі» – **ModelNotation**

Класи controllervalidation, modelnotation

Клас ControllerValidation

Клас ControllerValidation виконує валідацію введених користувачем даних для додавання або редагування сільськогосподарського підприємства.

**Основні функції класу:**

1. Перевіряє, чи заповнені всі обов'язкові поля (Name, OwnershipType, LandArea, EmployeesCount).
2. Перевіряє, чи тип власності належить до допустимих значень ("Д", "П", "К").
3. Перевіряє, чи площа землі більше нуля.
4. Перевіряє, чи кількість працівників не менше одного.
5. Якщо усі дані валідні — зберігає підприємство у базу даних.
6. При наявності помилок — повертає користувача на форму з повідомленнями про помилки.

using Microsoft.AspNetCore.Mvc;

using Task\_1.Models;

namespace Task\_1.Controllers

{

public class HomeController : Controller

{

private readonly ApplicationDbContext \_context;

public HomeController(ApplicationDbContext context)

{

\_context = context;

}

[HttpGet]

public IActionResult Index()

{

var enterprises = \_context.AgriculturalEnterprises.ToList();

return View(new EnterpriseFormViewModel

{

Enterprises = enterprises,

NewEnterprise = new AgriculturalEnterprise()

});

}

[HttpPost]

public IActionResult Index(EnterpriseFormViewModel model)

{

var newEnterprise = model.NewEnterprise;

if (string.IsNullOrWhiteSpace(newEnterprise.Name))

ModelState.AddModelError("NewEnterprise.Name", "Назва обов'язкова");

if (string.IsNullOrWhiteSpace(newEnterprise.OwnershipType) ||

!"ДПК".Contains(newEnterprise.OwnershipType))

ModelState.AddModelError("NewEnterprise.OwnershipType", "Недопустимий тип власності");

if (newEnterprise.LandArea <= 0)

ModelState.AddModelError("NewEnterprise.LandArea", "Площа повинна бути більшою за 0");

if (newEnterprise.EmployeesCount < 1)

ModelState.AddModelError("NewEnterprise.EmployeesCount", "Кількість працівників повинна бути не менше 1");

if (ModelState.IsValid)

{

\_context.AgriculturalEnterprises.Add(newEnterprise);

\_context.SaveChanges();

return RedirectToAction("Index");

}

model.Enterprises = \_context.AgriculturalEnterprises.ToList();

return View(model);

}

[HttpPost]

public IActionResult DeleteAll()

{

\_context.AgriculturalEnterprises.RemoveRange(\_context.AgriculturalEnterprises);

\_context.SaveChanges();

return RedirectToAction("Index");

}

[HttpPost]

public IActionResult Delete(int id)

{

var enterprise = \_context.AgriculturalEnterprises.Find(id);

if (enterprise != null)

{

\_context.AgriculturalEnterprises.Remove(enterprise);

\_context.SaveChanges();

}

return RedirectToAction("Index");

}

[HttpGet]

public IActionResult Edit(int id)

{

var enterprise = \_context.AgriculturalEnterprises.Find(id);

if (enterprise == null)

return NotFound();

return View(enterprise);

}

[HttpPost]

public IActionResult Edit(AgriculturalEnterprise model)

{

// Валідація через контролер

if (string.IsNullOrWhiteSpace(model.Name))

ModelState.AddModelError("Name", "Назва обов'язкова");

if (string.IsNullOrWhiteSpace(model.OwnershipType) ||

!"ДПК".Contains(model.OwnershipType))

ModelState.AddModelError("OwnershipType", "Недопустимий тип власності");

if (model.LandArea <= 0)

ModelState.AddModelError("LandArea", "Площа повинна бути більшою за 0");

if (model.EmployeesCount < 1)

ModelState.AddModelError("EmployeesCount", "Кількість працівників повинна бути не менше 1");

if (ModelState.IsValid)

{

\_context.AgriculturalEnterprises.Update(model);

\_context.SaveChanges();

return RedirectToAction("Index");

}

return View(model);

}

}

}

using Microsoft.AspNetCore.Mvc;

using Task\_1\_2.Models;

namespace Task\_1\_2.Controllers

{

public class HomeController : Controller

{

private readonly ApplicationDbContext \_context;

public HomeController(ApplicationDbContext context)

{

\_context = context;

}

[HttpGet]

public IActionResult Index()

{

var enterprises = \_context.AgriculturalEnterprises.ToList();

return View(new EnterpriseFormViewModel { Enterprises = enterprises, NewEnterprise = new AgriculturalEnterprise() });

}

[HttpPost]

public IActionResult Index(EnterpriseFormViewModel model)

{

// Валідація через контролер

if (string.IsNullOrWhiteSpace(model.NewEnterprise.Name))

ModelState.AddModelError("NewEnterprise.Name", "Назва є обов’язковою");

if (string.IsNullOrWhiteSpace(model.NewEnterprise.OwnershipType))

ModelState.AddModelError("NewEnterprise.OwnershipType", "Вид власності є обов’язковим");

if (model.NewEnterprise.LandArea <= 0)

ModelState.AddModelError("NewEnterprise.LandArea", "Площа має бути більшою за 0");

if (model.NewEnterprise.EmployeesCount < 1)

ModelState.AddModelError("NewEnterprise.EmployeesCount", "Кількість працівників повинна бути не менше 1");

if (string.IsNullOrWhiteSpace(model.NewEnterprise.ProductType))

ModelState.AddModelError("NewEnterprise.ProductType", "Тип продукції є обов’язковим");

if (ModelState.IsValid)

{

\_context.AgriculturalEnterprises.Add(model.NewEnterprise);

\_context.SaveChanges();

return RedirectToAction("Index");

}

var enterprises = \_context.AgriculturalEnterprises.ToList();

return View(new EnterpriseFormViewModel { Enterprises = enterprises, NewEnterprise = model.NewEnterprise });

}

[HttpPost]

public IActionResult DeleteAll()

{

\_context.AgriculturalEnterprises.RemoveRange(\_context.AgriculturalEnterprises);

\_context.SaveChanges();

return RedirectToAction("Index");

}

[HttpPost]

public IActionResult Delete(int id)

{

var enterprise = \_context.AgriculturalEnterprises.Find(id);

if (enterprise != null)

{

\_context.AgriculturalEnterprises.Remove(enterprise);

\_context.SaveChanges();

}

return RedirectToAction("Index");

}

[HttpGet]

public IActionResult Edit(int id)

{

var enterprise = \_context.AgriculturalEnterprises.Find(id);

if (enterprise == null)

{

return NotFound();

}

return View(enterprise);

}

[HttpPost]

public IActionResult Edit(AgriculturalEnterprise model)

{

// Валідація через контролер

if (string.IsNullOrWhiteSpace(model.Name))

ModelState.AddModelError("Name", "Назва є обов’язковою");

if (string.IsNullOrWhiteSpace(model.OwnershipType))

ModelState.AddModelError("OwnershipType", "Вид власності є обов’язковим");

if (model.LandArea <= 0)

ModelState.AddModelError("LandArea", "Площа має бути більшою за 0");

if (model.EmployeesCount < 1)

ModelState.AddModelError("EmployeesCount", "Кількість працівників повинна бути не менше 1");

if (string.IsNullOrWhiteSpace(model.ProductType))

ModelState.AddModelError("ProductType", "Тип продукції є обов’язковим");

if (ModelState.IsValid)

{

\_context.AgriculturalEnterprises.Update(model);

\_context.SaveChanges();

return RedirectToAction("Index");

}

return View(model);

}

}

}

@model Task\_1.Models.EnterpriseFormViewModel

@{

ViewData["Title"] = "Додавання підприємства";

}

<h2 class="text-center">Введіть дані підприємства</h2>

<div class="d-flex justify-content-center">

<**form** method="post" class="mt-4 p-4 bg-light shadow rounded" style="width: 400px;">

<div class="mb-3">

<label class="form-label">Назва:</label>

<**span** **asp-validation-for**="NewEnterprise.Name" class="text-danger small d-block"></**span**>

<**input** **asp-for**="NewEnterprise.Name" class="form-control" />

</div>

<div class="mb-3">

<label class="form-label">Вид власності:</label>

<**span** **asp-validation-for**="NewEnterprise.OwnershipType" class="text-danger small d-block"></**span**>

<**select** **asp-for**="NewEnterprise.OwnershipType" class="form-select">

<**option** **value**="">-- Оберіть --</**option**>

<**option** **value**="Д">Державна</**option**>

<**option** **value**="П">Приватна</**option**>

<**option** **value**="К">Кооперативна</**option**>

</**select**>

</div>

<div class="mb-3">

<label class="form-label">Площа землі (га):</label>

<**span** **asp-validation-for**="NewEnterprise.LandArea" class="text-danger small d-block"></**span**>

<**input** **asp-for**="NewEnterprise.LandArea" **type**="number" class="form-control" />

</div>

<div class="mb-3">

<label class="form-label">Кількість працівників:</label>

<**span** **asp-validation-for**="NewEnterprise.EmployeesCount" class="text-danger small d-block"></**span**>

<**input** **asp-for**="NewEnterprise.EmployeesCount" **type**="number" class="form-control" />

</div>

<div class="d-grid gap-2">

<button type="submit" class="btn btn-primary">Додати</button>

</div>

</**form**>

</div>

@if (Model.Enterprises.Any())

{

<h2 class="text-center mt-4">Список підприємств</h2>

<table class="table table-bordered mt-3">

<thead>

<tr>

<th>Назва</th>

<th>Вид власності</th>

<th>Площа землі (га)</th>

<th>Кількість працівників</th>

<th>Дії</th>

</tr>

</thead>

<tbody>

@foreach (var enterprise in Model.Enterprises)

{

<tr>

<td>@enterprise.Name</td>

<td>@enterprise.OwnershipType</td>

<td>@enterprise.LandArea</td>

<td>@enterprise.EmployeesCount</td>

<td>

<**a** **asp-action**="Edit" **asp-route-id**="@enterprise.Id" class="btn btn-sm btn-warning">Редагувати</**a**>

<**form** method="post" **asp-action**="Delete" style="display: inline-block;">

<input type="hidden" name="id" value="@enterprise.Id" />

<button type="submit" class="btn btn-sm btn-danger">Видалити</button>

</**form**>

</td>

</tr>

}

</tbody>

</table>

<div class="d-flex justify-content-start mt-2">

<**form** method="post" **asp-action**="DeleteAll">

<button type="submit" class="btn btn-danger">Видалити всі</button>

</**form**>

</div>

}

@model Task\_1.Models.AgriculturalEnterprise

@{

ViewData["Title"] = "Редагування підприємства";

}

<h2 class="text-center">Редагувати підприємство</h2>

<div class="d-flex justify-content-center">

<**form** method="post" class="mt-4 p-4 bg-light shadow rounded" style="width: 400px;">

<**input** **type**="hidden" **asp-for**="Id" />

<div class="mb-3">

<label class="form-label">Назва:</label>

<**span** **asp-validation-for**="Name" class="text-danger small d-block"></**span**>

<**input** **asp-for**="Name" class="form-control" />

</div>

<div class="mb-3">

<label class="form-label">Вид власності:</label>

<**span** **asp-validation-for**="OwnershipType" class="text-danger small d-block"></**span**>

<**select** **asp-for**="OwnershipType" class="form-select">

<**option** **value**="">-- Оберіть --</**option**>

<**option** **value**="Д">Державна</**option**>

<**option** **value**="П">Приватна</**option**>

<**option** **value**="К">Кооперативна</**option**>

</**select**>

</div>

<div class="mb-3">

<label class="form-label">Площа землі (га):</label>

<**span** **asp-validation-for**="LandArea" class="text-danger small d-block"></**span**>

<**input** **asp-for**="LandArea" **type**="number" class="form-control" />

</div>

<div class="mb-3">

<label class="form-label">Кількість працівників:</label>

<**span** **asp-validation-for**="EmployeesCount" class="text-danger small d-block"></**span**>

<**input** **asp-for**="EmployeesCount" **type**="number" class="form-control" />

</div>

<div class="d-grid">

<button type="submit" class="btn btn-success">Зберегти зміни</button>

<**a** **asp-action**="Index" class="btn btn-secondary mt-2">Скасувати</**a**>

</div>

</**form**>

</div>

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунки 7.1-7.4 – Результат роботи класу ControllerValidation

Клас ModelNotation

Клас ModelNotation описує модель даних для збереження інформації про сільськогосподарське підприємство в базі даних.

**Основні функції класу:**

1. Визначає таблицю AgriculturalEnterprise у базі даних.
2. Встановлює Id як первинний ключ із автогенерацією.
3. Забезпечує валідацію полів моделі за допомогою атрибутів:
   * Required — обов'язковість полів.
   * Range — допустимий діапазон числових значень.
4. Містить властивості для збереження назви, типу власності, площі землі, кількості працівників та типу продукції.

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace Task\_2.Models

{

[Table("AgriculturalEnterprise")]

public class AgriculturalEnterprise

{

[Key]

public int Id { get; set; }

[Required(ErrorMessage = "Назва є обов’язковою.")]

public string Name { get; set; } = string.Empty;

[Required(ErrorMessage = "Оберіть тип власності.")]

public string OwnershipType { get; set; } = string.Empty;

[Range(1, int.MaxValue, ErrorMessage = "Площа має бути більшою за 0.")]

public int LandArea { get; set; }

[Range(1, int.MaxValue, ErrorMessage = "Кількість працівників має бути більшою за 0.")]

public int EmployeesCount { get; set; }

}

}

using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace Task\_2\_2.Models

{

[Table("AgriculturalEnterprise")]

public class AgriculturalEnterprise

{

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.Identity)]

public int Id { get; set; }

[Required(ErrorMessage = "Назва обов'язкова")]

public string Name { get; set; } = string.Empty;

[Required(ErrorMessage = "Тип власності обов'язковий")]

public string OwnershipType { get; set; } = string.Empty;

[Range(1, int.MaxValue, ErrorMessage = "Площа має бути більше 0")]

public int LandArea { get; set; }

[Range(1, int.MaxValue, ErrorMessage = "Кількість працівників має бути більше 0")]

public int EmployeesCount { get; set; }

[Required(ErrorMessage = "Тип продукції обов'язковий")]

public string ProductType { get; set; } = string.Empty;

}

}

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунки 7.5-7.8 – Результат роботи класу ModelNotation

Висновок

У ході виконання цієї лабораторної роботи я ознайомився з використанням засобів валідації (перевірки на коректність) інформації під час її обробки у MVC WEB-застосунку. Було розглянуто різні способи перевірки правильності введених даних у формах при їх додаванні, редагуванні та збереженні у базі даних.

У процесі виконання роботи я реалізував наступні компоненти: – AgriculturalEnterprise – модель, що містить атрибути валідації для перевірки коректності введених даних: обов'язковість заповнення полів, діапазон допустимих значень, а також відображення повідомлень про помилки при неправильному вводі. – Контролери та представлення – реалізовано перевірку введених користувачем даних у формах та відображення відповідних повідомлень при помилках введення. – Перевірка на рівні БД – забезпечено збереження лише валідних даних за допомогою атрибутів DataAnnotations.

Ця робота допомогла мені закріпити практичні навички роботи з механізмами валідації у MVC WEB-застосунках та навчитися забезпечувати надійність і коректність введення та збереження даних користувачем. Отримані знання є важливими для подальшої розробки якісних та безпечних веб-додатків.