

ЛАБОРАТОРНА РОБОТА № 4

CRUD-операції в EF Core. Валідація даних.

Мета: набути навичок роботи з ORM, навчитися реалізовувати операції створення, читання, оновлення та видалення даних з використанням EF Core, набути навичок роботи з механізмами валідації даних.

Хід роботи:

Завдання 1.

Використовуючи підхід Code First, створити модель, яка разом з існуючою моделлю, утворюватиме зв'язок один-до-багатьох.

Лістининг Booking.cs:

```
public class Booking : IValidatableObject
{
    public long BookingID { get; set; }

    [Required(ErrorMessage = "Вкажіть ім'я гостя")]
    [StringLength(100, ErrorMessage = "Ім'я занадто довге")]
    public string GuestName { get; set; } = string.Empty;

    [Required(ErrorMessage = "Вкажіть дату заїзду")]
    public DateTime DateFrom { get; set; }

    [Required(ErrorMessage = "Вкажіть дату виїзду")]
    public DateTime DateTo { get; set; }

    [Required]
    public long RoomID { get; set; }

    public Room? Room { get; set; }
    public IEnumerable<ValidationResult> Validate(ValidationContext validationContext)
    {
        if (DateFrom >= DateTo)
        {
            yield return new ValidationResult(
                "Дата виїзду повинна бути пізніше дати заїзду.",
                new[] { nameof(DateTo) });
        }
    }
}
```

Завдання 2. Навігація по категоріям

Реалізуйте навігацію за категорією (або за іншою ознакою, яка більше підходить за вашим варіантом), дотримуючись наступних кроків:

					ДУ «Житомирська політехніка».25.121.06.000 –Лр4			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Вонсович Д.Б.			Звіт з лабораторної роботи №4		Літ.	Арк.
Перевір.		Українець М.О.						1
Керівник								14
Н. контр.							ФІКТ Гр. ІПЗ-23-2[1]	
Зав. каф.								

1. В папці проекту створіть папку Components, в ній створіть файл з назвою `NavigationViewComponent.cs` та реалізуйте його наступним чином:
В загальному шаблоні (файл `_Layout.cshtml`) додайте розмітку для сайд-бару. В доданій розмітці додайте виклик створеного `NavigationViewComponent`. Структура вашої сторінки має бути схожою на ту, яка зображена на рисунку 2.1.
3. Змініть реалізацію `NavigationViewComponent` таким чином, щоб компонент отримував перелік категорій з БД і повертав результат типу `IViewComponentResult`.
4. В папці `Views/Shared/Components/NavigationView` створіть файл `Default.cshtml`. Детальніше про конвенції неймінгу, які використовує Razor для роботи з ViewComponents можна [почитати тут](#).
5. В створеному файлі представлення реалізуйте виведення списку посилань, отриманих з ViewComponent'a.
6. У файлі `Controllers/HomeController.cs` в методі `Index()` реалізуйте підтримку фільтрації по категоріям.
7. Змініть `ViewModel` таким чином, щоб мати інформацію про обрану категорію на представленні. Це знадобиться для формування посилань на сторінки з урахуванням обраної категорії.
8. Реалізуйте виділення категорії, обраної користувачем. Для цього передавайте інформацію про обрану категорію з View Component в View.
9. Змініть реалізацію пейджингу таким чином, щоб він відображав кількість доступних сторінок відповідно до обраної категорії.

Лістининг `EFBookingRepository.cs`:

```
public class EFBookingRepository : IBookingRepository
{
    private BookingDbContext context;
    public EFBookingRepository(BookingDbContext ctx)
    {
        context = ctx;
    }
    public IQueryable<Room> Rooms => context.Rooms;
    public IQueryable<Booking> Bookings => context.Bookings;
```

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Пр4	Арк.
		Українець М.О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public void CreateRoom(Room room)
{
    context.Rooms.Add(room);
    context.SaveChanges();
}

public void UpdateRoom(Room room)
{
    context.Rooms.Update(room);
    context.SaveChanges();
}

public void DeleteRoom(Room room)
{
    context.Rooms.Remove(room);
    context.SaveChanges();
}

public void CreateBooking(Booking booking)
{
    context.Bookings.Add(booking);
    context.SaveChanges();
}

public void UpdateBooking(Booking booking)
{
    context.Bookings.Update(booking);
    context.SaveChanges();
}

public void DeleteBooking(Booking booking)
{
    context.Bookings.Remove(booking);
    context.SaveChanges();
}

```

Лістинг HomeController.cs:

```

public class HomeController : Controller
{
    private IBookingRepository repository;

    public HomeController(IBookingRepository repo)
    {
        repository = repo;
    }

    public IActionResult Index(string roomClass, int page = 1)
    {
        var filteredRooms = repository.Rooms
            .Where(r => roomClass == null || r.Class == roomClass);

        int pageSize = 2;

        var model = new ListViewModel
        {
            Rooms = filteredRooms
                .OrderBy(r => r.RoomID)
                .Skip((page - 1) * pageSize)
                .Take(pageSize),
            PagingInfo = new PagingInfo
            {
                CurrentPage = page,
                ItemsPerPage = pageSize,
            }
        }
    }
}

```

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Лр4	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

        TotalItems = filteredRooms.Count()
    }
};

    return View(model);
}
public IActionResult Details(long id)
{
    var room = repository.Rooms.FirstOrDefault(r => r.RoomID == id);
    if (room == null) return NotFound();
    return View(room);
}

public IActionResult Create()
{
    return View();
}

[HttpPost]
public IActionResult Create(Room room)
{
    if (ModelState.IsValid)
    {
        repository.CreateRoom(room);
        return RedirectToAction("Index");
    }
    return View(room);
}

public IActionResult Edit(long id)
{
    var room = repository.Rooms.FirstOrDefault(r => r.RoomID == id);
    if (room == null) return NotFound();
    return View(room);
}

[HttpPost]
[HttpPost]
public IActionResult Edit(Room room)
{
    if (ModelState.IsValid)
    {
        var existingRoom = repository.Rooms.FirstOrDefault(r => r.RoomID ==
room.RoomID);
        if (existingRoom != null)
        {
            existingRoom.Number = room.Number;
            existingRoom.Description = room.Description;
            existingRoom.Price = room.Price;
            existingRoom.Class = room.Class;
            repository.UpdateRoom(existingRoom);
        }
        return RedirectToAction("Index");
    }

    return View(room);
}

public IActionResult Delete(long id)
{
    var room = repository.Rooms.FirstOrDefault(r => r.RoomID == id);
    if (room == null) return NotFound();
    return View(room);
}
}

```

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Пр4	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

[HttpPost, ActionName("Delete")]
public IActionResult DeleteConfirmed(long id)
{
    var room = repository.Rooms.FirstOrDefault(r => r.RoomID == id);
    if (room != null)
    {
        repository.DeleteRoom(room);
    }
    return RedirectToAction("Index");
}
}

```

Лістининг BookingController.cs:

```

public class BookingController : Controller
{
    private readonly IBookingRepository repository;

    public BookingController(IBookingRepository repo)
    {
        repository = repo;
    }
    private const string SessionKey = "BookingInfo";

    [HttpPost]
    public IActionResult AddToBooking(int roomId, string guestName, DateTime dateFrom,
    DateTime dateTo)
    {
        var booking = HttpContext.Session.GetObject<List<BookingInfo>>(SessionKey) ?? new
    List<BookingInfo>();

        booking.Add(new BookingInfo
        {
            RoomID = roomId,
            GuestName = guestName,
            DateFrom = dateFrom,
            DateTo = dateTo
        });

        HttpContext.Session.SetObject(SessionKey, booking);
        return RedirectToAction("ShowBooking");
    }

    public IActionResult ShowBooking()
    {
        var booking = HttpContext.Session.GetObject<List<BookingInfo>>(SessionKey) ?? new
    List<BookingInfo>();
        return View(booking);
    }

    public IActionResult Pay()
    {
        var booking = HttpContext.Session.GetObject<List<BookingInfo>>(SessionKey);
        HttpContext.Session.Remove(SessionKey);
        return RedirectToAction("Index", "Home");
    }

    public IActionResult ClearBooking()
    {
        HttpContext.Session.Remove(SessionKey);
        return RedirectToAction("Index", "Home");
    }

    public IActionResult Index()
    {

```

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Пр4	Арк.
		Українець М.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        var bookings = repository.Bookings.Include(b => b.Room).ToList();
        return View(bookings);
    }

    public IActionResult Details(long id)
    {
        var booking = repository.Bookings.FirstOrDefault(b => b.BookingID == id);
        if (booking == null) return NotFound();
        return View(booking);
    }

    public IActionResult Create()
    {
        ViewBag.Rooms = repository.Rooms.ToList();
        return View();
    }

    [HttpPost]
    public IActionResult Create(Booking booking)
    {
        if (ModelState.IsValid)
        {
            bool isOverlapping = repository.Bookings.Any(b =>
                b.RoomID == booking.RoomID &&
                (
                    (booking.DateFrom >= b.DateFrom && booking.DateFrom < b.DateTo) ||
                    (booking.DateTo > b.DateFrom && booking.DateTo <= b.DateTo) ||
                    (booking.DateFrom <= b.DateFrom && booking.DateTo >= b.DateTo)
                )
            );

            if (isOverlapping)
            {
                ModelState.AddModelError("", "Ця кімната вже заброньована на вибрані  
дати.");
            }
            else
            {
                repository.CreateBooking(booking);
                return RedirectToAction("Index");
            }
        }

        ViewBag.Rooms = repository.Rooms.ToList();
        return View(booking);
    }

    public IActionResult Edit(long id)
    {
        var booking = repository.Bookings.FirstOrDefault(b => b.BookingID == id);
        if (booking == null) return NotFound();
        ViewBag.Rooms = repository.Rooms.ToList();
        return View(booking);
    }

    [HttpPost]
    public IActionResult Edit(Booking booking)
    {
        if (ModelState.IsValid)
        {
            var existingBooking = repository.Bookings.FirstOrDefault(b => b.BookingID ==
            booking.BookingID);
            if (existingBooking != null)
            {
                existingBooking.GuestName = booking.GuestName;
            }
        }
    }

```

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Пр4	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

        existingBooking.DateFrom = booking.DateFrom;
        existingBooking.DateTo = booking.DateTo;
        existingBooking.RoomID = booking.RoomID;
        repository.UpdateBooking(existingBooking);
    }
    return RedirectToAction("Index");
}
ViewBag.Rooms = repository.Rooms.ToList();
return View(booking);
}

public IActionResult Delete(long id)
{
    var booking = repository.Bookings.Include(b => b.Room).FirstOrDefault(b =>
b.BookingID == id);
    if (booking == null) return NotFound();
    return View(booking);
}

[HttpPost, ActionName("Delete")]
public IActionResult DeleteConfirmed(long id)
{
    var booking = repository.Bookings.FirstOrDefault(b => b.BookingID == id);
    if (booking != null)
    {
        repository.DeleteBooking(booking);
    }
    return RedirectToAction("Index");
}
}

```

Лістинг HomeController.cs:

```

using Microsoft.AspNetCore.Mvc;
using BookingHotel.Models;
using BookingHotel.Models.ViewModels;

namespace BookingHotel.Controllers
{
    public class HomeController : Controller
    {
        private IBookingRepository repository;

        public HomeController(IBookingRepository repo)
        {
            repository = repo;
        }

        public IActionResult Index(string roomClass, int page = 1)
        {
            var filteredRooms = repository.Rooms
                .Where(r => roomClass == null || r.Class == roomClass);

            int pageSize = 2;

            var model = new ListViewModel
            {
                Rooms = filteredRooms
                    .OrderBy(r => r.RoomID)
                    .Skip((page - 1) * pageSize)
                    .Take(pageSize),
                PagingInfo = new PagingInfo
                {
                    CurrentPage = page,
                    ItemsPerPage = pageSize,

```

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Пр4	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

```

        TotalItems = filteredRooms.Count()
    }
};

    return View(model);
}
}
}

```

Лістининг Views/Home/Index.cshtml:

```

@model BookingHotel.Models.ViewModels.ListViewModel

<div class="row">
    @foreach (var room in Model.Rooms ?? Enumerable.Empty<Room>())
    {
        <partial name="RoomSummary" model="room" />

        <a asp-controller="Booking" asp-action="Create" asp-route-
roomId="BookingHotel.Models.Room.RoomID" class="btn btn-success m-2">
            Забронювати окремо
        </a>
        <a asp-controller="Booking" asp-action="Index" class="btn btn-info m-2">
            Заброньовані номери
        </a>
        <a asp-controller="Home" asp-action="Create" class="btn btn-secondary m-2">
            Створити номер
        </a>
    }
</div>

<div class="pagination justify-content-center">
    @for (int i = 1; i <= Model.PagingInfo.TotalPages; i++)
    {
        <a href="@Url.Action("Index", new { page = i })"
class="page-link @(i == Model.PagingInfo.CurrentPage ? "active" : "")">@i</a>
    }
</div>

```

Лістининг Views/Home/Edit.cshtml:

```

@model BookingHotel.Models.Room

<h2>@ViewData["Title"]</h2>

<form asp-action="Edit">
    <input type="hidden" asp-for="RoomID" />
    <div class="form-group">
        <label asp-for="Number"></label>
        <input asp-for="Number" class="form-control" />
        <span asp-validation-for="Number" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="Description"></label>
        <input asp-for="Description" class="form-control" />
        <span asp-validation-for="Description" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="Price"></label>
        <input asp-for="Price" class="form-control" />
        <span asp-validation-for="Price" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="Class"></label>
        <input asp-for="Class" class="form-control" />
    </div>
</form>

```

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Лр4	Арк.
		Українець М.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        <span asp-validation-for="Class" class="text-danger"></span>
    </div>
    <button type="submit" class="btn btn-primary">Зберегти</button>
</form>

```

Лістининг Views/Home/Details.cshtml:

```

@model BookingHotel.Models.Room

<h2>Деталі номера</h2>

<div>
    <h4>Номер: @Model.Number</h4>
    <p>Опис: @Model.Description</p>
    <p>Ціна: @Model.Price</p>
    <p>Клас: @Model.Class</p>
</div>

<p>
    <a asp-action="Edit" asp-route-id="@Model.RoomID">Редагувати</a> |
    <a asp-action="Index">Повернутися до списку</a>
</p>

```

Лістининг Views/Booking/Index.cshtml:

```

@model List<BookingHotel.Models.Booking>

<h2>Список бронювань</h2>

<p>
    <a asp-action="Create" class="btn btn-primary">Додати нове бронювання</a>
</p>

<table class="table">
    <thead>
        <tr>
            <th>Ім'я гостя</th>
            <th>Дата заїзду</th>
            <th>Дата виїзду</th>
            <th>Номер</th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        @foreach (var booking in Model)
        {
            <tr>
                <td>@booking.GuestName</td>
                <td>@booking.DateFrom.ToShortDateString()</td>
                <td>@booking.DateTo.ToShortDateString()</td>
                <td>@booking.Room?.Number</td>
                <td>
                    <a asp-action="Edit" asp-route-id="@booking.BookingID">Редагувати</a>
                    |
                    <a asp-action="Details" asp-route-id="@booking.BookingID">Деталі</a>
                    |
                    <a asp-action="Delete" asp-route-id="@booking.BookingID">Видалити</a>
                </td>
            </tr>
        }
    </tbody>
</table>

<a asp-controller="Home" asp-action="Index" class="btn btn-success m-3">

```

На головну

Лістининг Views/Booking/Edit.cshtml:

@model BookingHotel.Models.Booking

<h2>@ViewData["Title"]</h2>

```
<form asp-action="Edit">
    <input type="hidden" asp-for="BookingID" />

    <div class="form-group">
        <label asp-for="GuestName"></label>
        <input asp-for="GuestName" class="form-control" />
        <span asp-validation-for="GuestName" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="DateFrom"></label>
        <input asp-for="DateFrom" type="date" class="form-control" />
        <span asp-validation-for="DateFrom" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="DateTo"></label>
        <input asp-for="DateTo" type="date" class="form-control" />
        <span asp-validation-for="DateTo" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="RoomID">Номер</label>
        <select asp-for="RoomID" class="form-control" asp-items="@((new
SelectList(ViewBag.Rooms, "RoomID", "Number")))"></select>
        <span asp-validation-for="RoomID" class="text-danger"></span>
    </div>
    <button type="submit" class="btn btn-primary">Зберегти</button>
</form>
```

Лістининг Views/Booking/Edit.cshtml:

@model BookingHotel.Models.Booking

<h2>Видалити бронювання</h2>

<h3>Ви впевнені, що хочете видалити це бронювання?</h3>

```
<div>
    <h4>Ім'я гостя: @Model.GuestName</h4>
    <p>Дата заїзду: @Model.DateFrom.ToShortDateString()</p>
    <p>Дата виїзду: @Model.DateTo.ToShortDateString()</p>
    <p>Номер: @Model.Room?.Number</p>
</div>
```

```
<form asp-action="Delete" method="post">
    <input type="hidden" asp-for="BookingID" />
    <button type="submit" class="btn btn-danger">Видалити</button> |
    <a asp-action="Index">Скасувати</a>
</form>
```

Результат виконання:

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Пр4	Арк.
		Українець М.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

BookingHotel

Home

Rooms

Cheap

Lux

Normal

Super Lux

101

Broken shower

Lux

160,00 ₴

Ім'я гостя

дд.мм.р

дд.мм.р

Забронювати

Редагувати

Деталі

Видалити

101a

No description

Normal

150,00 ₴

Ім'я гостя

дд.мм.р

дд.мм.р

Забронювати

Редагувати

Деталі

Видалити

Забронювати окремо

Заброньовані номери

Створити номер

1

2

3

© 2025 BookingHotel

Рис. 2. Результат виконання

BookingHotel

Home

Rooms

Cheap

Lux

Normal

Super Lux

Number

101

Description

Broken shower

Price

170

Class

Lux

Зберегти

© 2025 BookingHotel

Рис. 3. Результат виконання

101

Broken shower

Lux

170,00 ₴

Ім'я гостя

дд.мм.р

дд.мм.р

Забронювати

Редагувати

Деталі

Видалити

Рис. 4. Результат виконання

Деталі номера

Номер: 101a

Опис: No description

Ціна: 150,00

Клас: Normal

[Редагувати](#) | [Повернутися до списку](#)

Рис. 5. Результат виконання

Список бронювань

Додати нове бронювання

Ім'я гостя	Дата заїзду	Дата виїзду	Номер	
Даня	03.05.2025	11.05.2025	101	Редагувати Деталі Видалити
Даня	17.05.2025	23.05.2025	230	Редагувати Деталі Видалити
Петро	11.05.2025	23.05.2025	101	Редагувати Деталі Видалити
Даня	17.05.2025	23.05.2025	101	Редагувати Деталі Видалити
Даніл	30.05.2025	06.06.2025	230	Редагувати Деталі Видалити

На головну

Рис. 6. Результат виконання

Видалити бронювання

Ви впевнені, що хочете видалити це бронювання?

Ім'я гостя: Даніл

Дата заїзду: 30.05.2025

Дата виїзду: 06.06.2025

Номер: 230

[Видалити](#) | [Скасувати](#)

Рис. 7. Результат виконання

Список бронювань

Додати нове бронювання

Ім'я гостя	Дата заїзду	Дата виїзду	Номер	
Даня	03.05.2025	11.05.2025	101	Редагувати Деталі Видалити
Даня	17.05.2025	23.05.2025	230	Редагувати Деталі Видалити
Петро	11.05.2025	23.05.2025	101	Редагувати Деталі Видалити
Даня	17.05.2025	23.05.2025	101	Редагувати Деталі Видалити

На головну

Рис. 8. Результат виконання

Завдання 3.

Додати серверну валідацію для перевірки введених користувачем даних перед виконанням CRUD-операцій. Використати DataAnnotations для реалізації перевірки введених даних, приклад використання DataAnnotations для сутності Products наведено в лістингу нижче:

Лістинг Room.cs:

```
public class Room
{
    [Key]
    public long RoomID { get; set; }

    [Required(ErrorMessage = "Вкажіть номер кімнати")]
    [StringLength(10, ErrorMessage = "Номер кімнати занадто довгий")]
    public string Number { get; set; } = String.Empty;

    [Required(ErrorMessage = "Вкажіть опис кімнати")]
    [StringLength(500, ErrorMessage = "Опис занадто довгий")]
    public string Description { get; set; } = String.Empty;

    [Required(ErrorMessage = "Вкажіть ціну")]
    [Range(0.01, 100000, ErrorMessage = "Ціна має бути в діапазоні від 0.01 до 100000")]
    [Column(TypeName = "decimal(8, 2)")]
    public decimal Price { get; set; }

    [Required(ErrorMessage = "Вкажіть клас кімнати")]
    [StringLength(50, ErrorMessage = "Клас занадто довгий")]
    public string Class { get; set; } = String.Empty;

    public List<Booking> Bookings { get; set; } = new();
}
```

Резальтат виконання:

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Лр4	Арк.
		Українець М.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Number
301

Description

Вкажіть опис кімнати

Price
ddd

The value 'ddd' is not valid for Price.

Class
Lux

Зберегти

Рис. 9. Результат виконання

GuestName
Вадим

DateFrom
02.05.2025

DateTo
23.05.2025

Номер
101

- Ця кімната вже заброньована на вибрані дати.

Зберегти На головну

Рис. 10. Результат виконання

Висновок

Під час виконання лабораторної роботи я набув навичок роботи з ORM, навчився реалізовувати операції створення, читання, оновлення та видалення даних з використанням EF Core, набув навичок роботи з механізмами валідації даних.

		Вонсович Д.Б.			ДУ «Житомирська політехніка».25.121.06.000 – Лр4	Арк.
		Українець М.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		