

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ОСТРОЗЬКА АКАДЕМІЯ”
Кафедра економіко-математичного моделювання та інформаційних
технологій

ЗВІТ
Лабораторна робота №2
Основи роботи з ASP.NET Core MVC

здобувачів вищої освіти
першого (бакалаврського) рівня
третього року навчання групи
КН-21
спеціальності 122 Комп’ютерні
науки
ОПП «Комп’ютерні науки»

Лайтер Ярини Семенівни
Колоди Станіслава Геннадійовича
(прізвище, ім’я, по батькові)

База лабораторної - кафедра економіко-математичного моделювання
та інформаційних технологій

Керівник лабораторної від університету

(науковий ступінь, учене звання керівника)

(Власне ім’я та ПРІЗВИЩЕ)

Члени комісії:

_____ Юрій КЛЕБАН

1. Опис завдання	3
3. Опис класів у вигляді коду, з коментарями	4
Код інтерфейсу IEmailSender:	4
Код класу IEmailSender:	4
Частина коду класу ResponceController:	5
Частина коду класу AssignmentController:	5
Частина коду класу ComplaintController(customer):	6
4. Опис графічного інтерфейсу програми:	8
5. Висновок	9
6. Список використаних джерел:	10

Inspector

Мета: Метою лабораторної роботи є набуття вмінь з інтеграція дизайну, побудови архітектури рішення на [ASP.NET](#) Core, а також закріплення навичок колективної роботи.

1. Опис завдання

Основне завдання лабораторної роботи №1: реалізувати CRUD для усіх сутностей, додати управління користувачами, інтегрувати дизайн, реалізувати трирівневу архітектуру (або складнішу).

- Продовжуємо працювати у тих же репозиторіях.
- Кожну задачу виконує визначений учасник у окремій гілці, **Team Lead** робить **merge** у **main**.
- Обрати шаблон для інтеграції у проєкт.
- Інтегрувати дизайну проєкт.
- Реалізувати CRUD-операції для усіх сутностей.
- Додати можливість управління користувачами, розділи функції за ролями + вхід/реєстрація.
- Реалізувати як мінімум трирівневу архітектуру з організацією коду у різних збірках.
- Виконати додаткове завдання до Лаб 2. Щоб знайти додаткове завдання потрібно натиснути **OPEN** на назву проєкту. (Додати сповіщення на електронну пошту про відповідь на скаргу або про перенаправлення скарги іншому виконавцю.)
- Підготовка звіту.
- Публічний захист.

3. Опис класів у вигляді коду, з коментарями

Код інтерфейсу IEmailSender:

```
public interface IEmailSender
{
    Task SendEmailAsync(string email, string? subject, string? htmlMessage, int? orgID);
}
```

Лістинг 1.1. Код програми

Короткий опис коду:

Цей код визначає інтерфейс IEmailSender, який містить один метод SendEmailAsync. Метод призначений для асинхронної відправки електронних листів

Код класу IEmailSender:

```
public class EmailSender : IEmailSender
{
    public Task SendEmailAsync(string email, string? subject, string? htmlMessage, int? orgID)
    {
        var mail = "inspectorapp2023@gmail.com";
        subject = "Response from inspector.web";
        if (orgID != null)
        {
            htmlMessage = "You have received a new task from your company!" +
                "\r\nGo check the web site to see more." +
                "\r\n\r\nRegards," +
                "\r\nThe Inspector team";
        }
        else
        {
            htmlMessage = "You have received a response for your complaint!" +
                "\r\nGo check the web site to see more." +
                "\r\n\r\nRegards," +
                "\r\nThe Inspector team";
        }

        var client = new SmtpClient("smtp.outlook.com", 587)
        {
            EnableSsl = true,
            UseDefaultCredentials = false,
            Credentials = new NetworkCredential(mail, "Inspector123!")
        };

        return client.SendMailAsync(
            new MailMessage(from: mail,
                           to: email,
                           subject,
                           htmlMessage));
    }
}
```

Лістинг 1.2. Код програми

Короткий опис коду:

Цей код описує клас, який відповідає за відправку листів користувачам. Листи надходять від корпоративної пошти нашої програми.

Частина коду класу **ResponseController**:

```
....
[HttpPost]
public IActionResult Create(Response response, IFormFile? file)
{
    if (ModelState.IsValid)
    {
        _responseRepo.Add(response);
        _responseRepo.Save();
        //sending email for the user when the response is given
        var user = _userRepo.Get(u => u.Id == response.UserTakeId);
        SendResponseEmail(user.Email, user.OrganizationId);
        TempData["success"] = "Response and email sent successfully";
        Complaint obj = _complaintRepo.Get(u => u.Id == response.ComplaintId);
        obj.ResponseId = response.Id;
        obj.Status = "done";
        _complaintRepo.Save();

        return RedirectToAction("Index", "Complaint");
    }
    return View(response);
}

public async Task SendResponseEmail(string email, int? orgId)
{
    await _emailSender.SendEmailAsync(email, null, null, orgId);
}
....
```

Лістинг 1.3. Код програми

Короткий опис коду:

Цей код надає можливість відправляти листи на пошту та сповіщати користувача, що його скарга була розглянута та виконана (за допомогою асинхронної функції).

Частина коду класу **AssignmentController**:

```
[HttpPost]
public IActionResult Create(AssignmentVM assignmentVM)
{
    if (ModelState.IsValid)
    {
        _assignmentRepo.Add(assignmentVM.Assignment);

        var user = _userRepo.Get(u => u.Id ==
assignmentVM.Assignment.UserTakeId);
        SendResponseEmail(user.Email, user.OrganizationId);

        TempData["success"] = "Assignment created successfully";
        _assignmentRepo.Save();
    }
}
```

```

        Complaint complaint = _complaintRepo.Get(u => u.Id == assignmentVM.Assignment.ComplaintId);
        _complaintRepo.Get(u => u.Id == assignmentVM.Assignment.ComplaintId).Status = "in process";
        _complaintRepo.Save();

        return RedirectToAction("Index", "Complaint");
    }
    return View();
}

public async Task SendResponseEmail(string email, int? orgID)
{
    await _emailSender.SendEmailAsync(email, null, null, orgID);
}

```

Лістинг 1.4. Код програми

Короткий опис коду:

Цей код надає можливість відправляти листи на пошту робітникам, яким доручили виконувати роботу\скаргу (за допомогою асинхронної функції).

Частина коду класу **ComplaintController(customer)**:

```

[HttpPost]
public IActionResult Upsert(ComplaintVM complaintVM, List<IFormFile>? files)
{
    if (ModelState.IsValid)
    {
        string wwwRootPath = _webHostEnvironment.WebRootPath;

        if (complaintVM.Complaint.Id == 0)
        {
            complaintVM.Complaint.UserId = User.FindFirstValue(ClaimTypes.NameIdentifier);
            _complaintRepo.Add(complaintVM.Complaint);
            TempData["success"] = "Complaint created successfully";
        }
        else
        {
            complaintVM.Complaint.UserId = User.FindFirstValue(ClaimTypes.NameIdentifier);
            _complaintRepo.Update(complaintVM.Complaint);

            //видалення файлів які були збережені на цю скаргу раніше
            var complaintFiles = _complaintFileRepo.GetAll().Where(cf => cf.ComplaintId ==
complaintVM.Complaint.Id).ToList();

            foreach (var compFile in complaintFiles)
            {
                if (!string.IsNullOrEmpty(compFile.FilePath))
                {
                    var oldImagePath = Path.Combine(wwwRootPath, compFile.FilePath.TrimStart('/'));

                    if (System.IO.File.Exists(oldImagePath))
                    {
                        System.IO.File.Delete(oldImagePath);
                        _complaintFileRepo.Remove(compFile);
                        _complaintFileRepo.Save();
                    }
                }
            }
        }
    }
}

```

```

        TempData["success"] = "Complaint update successfully";
    }

    _complaintRepo.Save();

    if (files != null && files.Count > 0)
    {
        foreach (var file in files)
        {
            if (file != null && file.Length > 0)
            {
                string fileName = Guid.NewGuid().ToString() +
                    Path.GetExtension(file.FileName);
                string filePath = Path.Combine(wwwRootPath,
                    @"files");

                using (var fileStream = new FileStream(Path.Combine(filePath, fileName), FileMode.Create))
                {
                    file.CopyTo(fileStream);
                }

                var complaintFile = new ComplaintFile
                {
                    FileName = fileName,
                    FilePath = "/files/" + fileName,
                    ComplaintId = complaintVM.Complaint.Id
                };

                _complaintFileRepo.Add(complaintFile);
                _complaintFileRepo.Save();
            }
        }

        return RedirectToAction("Index");
    }
    else
    {
        complaintVM.OrganizationList = _organizationRepo.GetAll().Select(u => new SelectListItem
        {
            Text = u.Name,
            Value = u.Id.ToString()
        });

        return View(complaintVM);
    }
}

```

Лістинг 1.5. Код програми

Короткий опис коду:

Цей код надає можливість користувачу завантажувати декілька файлів у свою заявку. Якщо користувач вирішить редагувати заявку, всі попередні файли будуть видалені, а підкріплені до цієї заявки будуть тільки нові файли.

4. Опис графічного інтерфейсу програми:

Complaint List + Create Complaint

Show entries Search:

FullName	Organization	Description	Status	
Bob Smith	Health Organization	There are problem with helth	sent	Edit Delete
Jane Smith	Water Organization	There are problem with water	sent	Edit Delete
John Doe	Road Organization	There are problem with road	done	Edit Delete

Рис 1.1. Інтерфейс програми

Create

OrganizationId
--SelectOrganization--

Description

File Edit View Insert Format Tools Table

↶ ↷ Paragraph System Font 12pt B I U

p 0 words tiny

File

Вибрати файл Файл не вибрано

Create Back to List

Рис 1.2. Інтерфейс програми

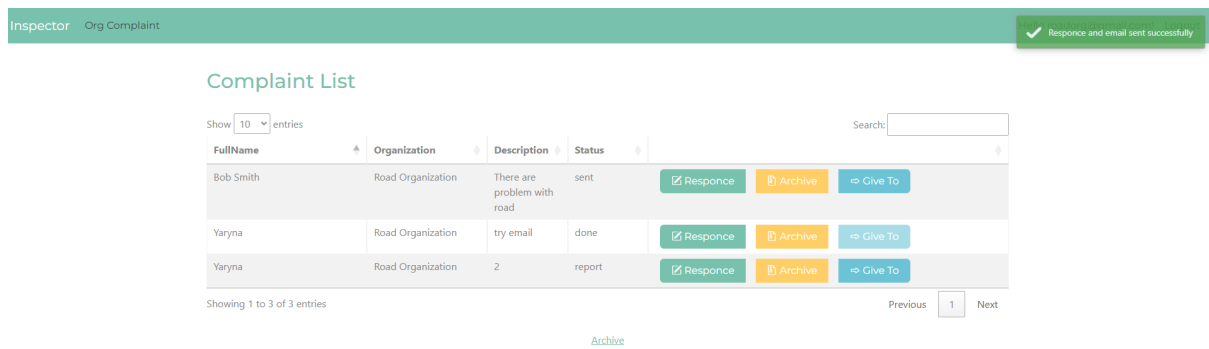



Рис 1.3. Інтерфейс програми



6. Список використаних джерел:

1. Звіт №1 -  Звіт_Лаб1_Лайтер_Колода
2. Репозиторій на github - <https://github.com/oa-ek/lab-1-inspector>
3. C# Guide - <https://docs.microsoft.com/en-us/dotnet/csharp/>.
4. C# Station - <https://csharp-station.com/>.
5. Working with C# - <https://code.visualstudio.com/docs/languages/csharp>.
6. Основи програмування на C# (Prometheus) ,
https://edx.prometheus.org.ua/courses/Microsoft/CS201/2016_T1/info
7. Introduction to C# (Edx), <https://www.edx.org/course/introduction-to-c>
8. DotNetMastery - <https://dotnetmastery.com/>