**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра інженерії програмного забезпечення**

**ЗВІТ**

*з лабораторної роботи № 10*

**«Валідація форм у ASP.NET Core»**

**Варіант № 11**

Дисципліна «розробка web-застосунків на базі ASP.NET»

Спеціальність «Інженерія програмного забезпечення»

121–ЛР.10–310. 22111012

***Cтудент*** *\_\_\_\_\_\_\_\_\_****А. М.******Карачебан***

*(підпис)*

*\_\_05.10.2023 \_\_*

*(дата)*

***Викладач*** *\_\_\_\_\_\_\_\_\_****Є. Д. Стоєв***

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Миколаїв – 2023**

**Лабораторна робота №10**

***Тема роботи:***Валідація форм у ASP.NET Core

***Завдання:***

1. Створіть форму реєстрації на консультацію. У формі мають бути поля:

− Ім'я прізвище

− Email

− Бажана дата консультації

Усі поля є обов'язковими для введення. Email має бути у відповідному форматі. Дата консультації має бути правильною датою, вона має бути в майбутньому і не попадати на вихідні дні. За наявності помилок, необхідно повідомити користувача про помилки.

1. Для форми з попереднього завдання додайте список, в якому можна буде вказати продукт, за яким клієнт хоче отримати консультацію. Значення: JavaScript, C#, Java, Python, Основи. При цьому консультація щодо продукту «Основи» не може проходити по понеділках, відповідна помилка повинна виводитися в інтерфейсі користувача.

***Виклик програми:***

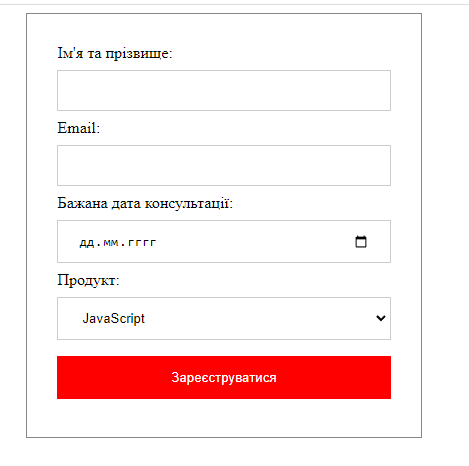


Рисунок 1 – Виклик програми(початкова форма)

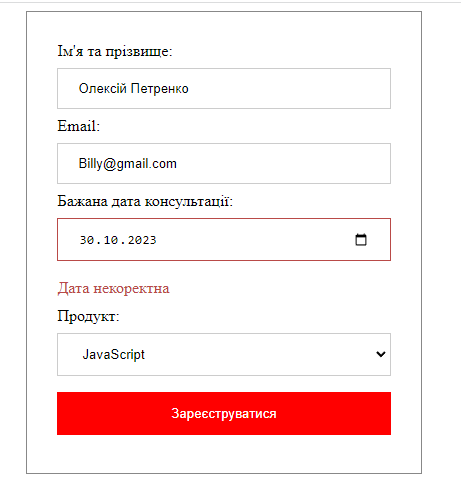


Рисунок 2 – Виклик програми(некоректна дата)

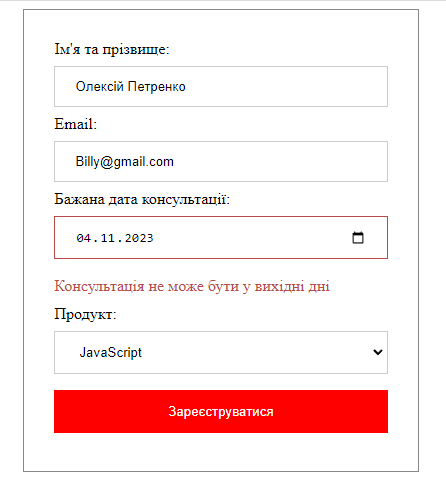


Рисунок 3 – Виклик програми(некоректна дата)

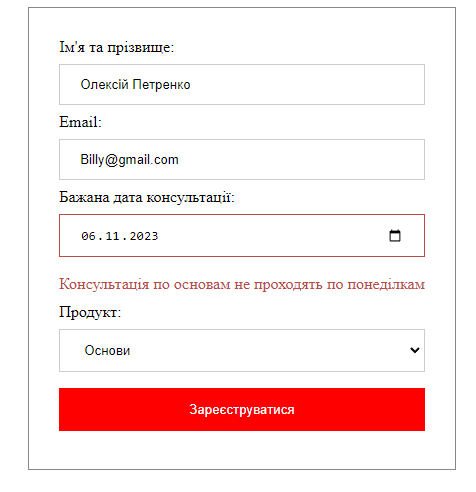


Рисунок 4 – Виклик програми(некоректна дата)

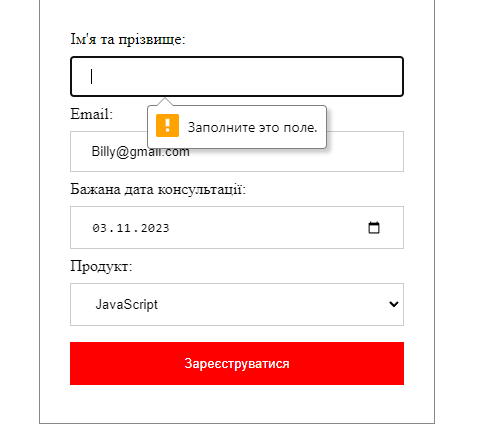


Рисунок 5 – Виклик програми(незаповнене поле)

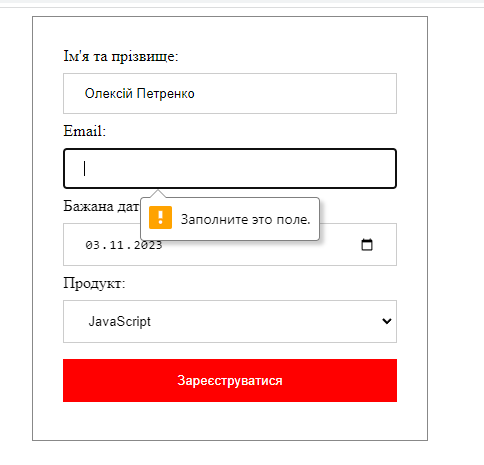


Рисунок 6 – Виклик програми(незаповнене поле)

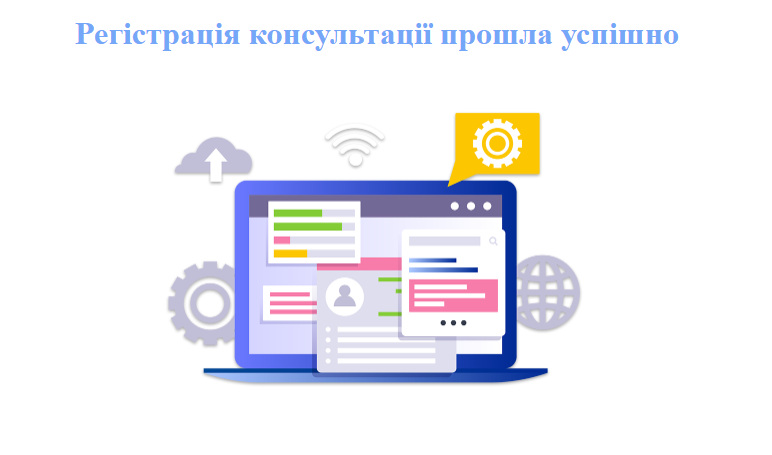


Рисунок 7 – Виклик програми(сторінка при успішній реєстрації)

***Висновки:*** Було досягнуто поставлену мету та розроблено програмний код. Для успішного виконання роботи було опрацьовано основи ASP. NET. Було вивчено що таке валідація у ASP.NET Core MVC. Було освоєно основи роботи з використання валідації у ASP.NET Core MVC.

***Посилання на Github:*** [***https://github.com/Lirirum/asp\_lab\_10***](https://github.com/Lirirum/asp_lab_10)

***Питання:***

1. ***Що таке дескрипторний допоміжний клас? Назвіть основні дескрипторні допоміжні класи та поясніть, як вони працюють***

Дескрипторні допоміжні класи використовуються для надання додаткових метаданих про типи даних в системі ASP.NET. Основна ідея полягає в тому, щоб надавати інформацію про те, як тип даних повинен відображатися та оброблятися на різних рівнях платформи.

***Основні дескрипторні допоміжні класи:***

* Display Attribute : використовується для визначення метаданих властивостей моделі (класу), які можуть бути використані для відображення цих властивостей у користувацькому інтерфейсі***.***
* DisplayName Attribute: Визначає людський читабельний текст, який повинен використовуватися для відображення імені властивості або поля.
* DisplayFormat Attribute: Задає формат відображення для поля чи властивості, такий як формат дати чи числа.
* DataType Attribute: Вказує, який тип даних слід використовувати для властивості чи поля (наприклад, текст, число, електронна адреса).
* StringLength Attribute: Встановлює максимальну та мінімальну довжину рядка для властивості.
* ScaffoldColumn Attribute: Атрибут ScaffoldColumn дозволяє приховати відображення властивості при використанні хелперів Html.DisplayForModel() та Html.EditorForModel().
* RegularExpression Attribute: Використання цього атрибуту передбачає, що значення, що вводиться, повинно відповідати зазначеному в цьому атрибуті регулярному виразу.
* Range Attribute: Вказує діапазон значень для числових полів.
* Required Attribute: Вказує, що певне поле у моделі (класі) повинно бути обов'язково заповненим перед тим, як дані можуть бути надіслані на сервер чи збережені в базі даних. Це особливо корисно при валідації форм та обробці даних.
* UIHint Atribute : вказує, який буде використовуватися шаблон відображення під час створення розмітки html для цієї властивості. Шаблон управляє, як властивість буде рендер на сторінці.

1. ***Що потрібно зробити для того, щоб у поданні почали працювати допоміжні класи?***

Для того, щоб у представленні почали працювати допоміжні класи, потрібно включити їх у властивості моделі. Наприклад, при оголошенні властивостей в класі моделі можна додавати атрибути, які вказують на різні метадані.

public class MyModel

{

[Required]

[StringLength(50)]

[Display(Name = "Ім'я")]

public string Name { get; set; }

// Інші властивості та атрибути

}

1. ***Що таке валідація даних? У чому різниця між валідацією на сервері та на клієнті?***

Валідація даних – це процес перевірки введених даних на відповідність певним правилам чи обмеженням. Різниця між валідацією на сервері та на клієнті полягає в тому, де саме відбувається цей процес:

* ***Валідація на сервері:*** Перевірка відбувається на серверній стороні, що забезпечує більш високий рівень безпеки та надійності. Забезпечується, що всі дані, які приходять від користувача, відповідають визначеним правилам.
* ***Валідація на клієнті:*** Перевірка відбувається на браузері користувача перед відправленням даних на сервер. Це може полегшити взаємодію з користувачем, але менш надійно, оскільки дані можуть бути обійдені або змінені.

1. ***За що відповідає атрибут UIHint?***

Атрибут UIHint вказує, який шаблон використовувати для відображення певного поля в інтерфейсі користувача. Він надає вказівку про те, як властивість або поле повинно бути відображене в інтерфейсі.

Основна ідея полягає в тому, щоб надати інформацію про те, як вивести певне поле на сторінці, вказавши ім'я шаблону, яке повинно бути використане. Наприклад, якщо у вас є модель з полем типу дати, і ви хочете використовувати свій власний шаблон для вибору дати, ви можете використовувати атрибут UIHint, щоб вказати ім'я шаблону, яке повинно бути використане для цього поля.

1. ***Що потрібно зробити для виконання валідації на стороні клієнта?***

Для виконання валідації на стороні клієнта можна використовувати JavaScript-бібліотеки, такі як jQuery або вбудовані можливості браузера(використання HTML5 атрибутів для перевірки: використовувати вбудовані можливості HTML5 для валідації, такі як required, min, max, pattern тощо.). Для включення валідації на стороні клієнта потрібно додати певні скрипти та налаштування, наприклад, використовуючи бібліотеку валідації jQuery або використовуючи вбудовані можливості в ASP.NET MVC або ASP.NET Core.