МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГИЙ УПРАВЛІННЯ

Лабораторна робота № 5

з курсу «Стек технологій .Net»

«Робота з даними в ASP.NET 2.0»

Виконав:

студент групи КН 36-а

Кулик В.В.

Перевірила:

Літвінова Ю.С.

ХАРКІВ 2018

**Мета роботи:**

Ознайомитись з рядом застосунків ASP.NET 2.0 для подання та відображення даних.

**Хід роботи:**

Варіант індивідуального завдання: №8 – Телефонна станція.

Опис механізму роботи розробленого додатку зі сховищем даних та коментарями по тексту програми:

Щоб відобразити данні про дзвінок на нашому сайті, необхідно створити сутність «Call Descriptor» на боці сервера, що представлена у вигляді:

public class CallDescriptor

{

private ConnectionDevice caller;

private ConnectionDevice called;

private bool isActive;

private int duration;

private DateTime start;

public CallDescriptor(ConnectionDevice caller, ConnectionDevice called, bool isActive, int duration, DateTime start)

{

this.caller = caller;

this.called = called;

this.isActive = isActive;

this.duration = duration;

this.start = start;

}

public ConnectionDevice Caller { get => caller; set => caller = value; }

public ConnectionDevice Called { get => called; set => called = value; }

public bool IsActive { get => isActive; set => isActive = value; }

public int Duration { get => duration; set => duration = value; }

public DateTime Start { get => start; set => start = value; }

Рисунок 1 – Програмний код сутності «Call Descriptor»

Як можна побачити, сутність «Call Descriptor» має 5 властивостей: Той, хто телефонує, Той, хто викликається, Активність, Тривалість, Початок дзвінка.

Уся множина робітника представлена збірним динамічним списком, де кожний елемент списку має приведений вище тип даних:

public class HomeController : Controller

{

public IActionResult Index()

{

CallDescriptor dsr1 = new CallDescriptor(new ConnectionDevice("Dima", "3458999"), new ConnectionDevice("Igor", "3458996"), false, 240, new DateTime(2018, 12, 10, 19, 22, 05));

CallDescriptor dsr2 = new CallDescriptor(new ConnectionDevice("Roman", "3458997"), new ConnectionDevice("Igor", "3458996"), false, 320, new DateTime(2018, 08, 10, 21, 20, 18));

CallDescriptor dsr3 = new CallDescriptor(new ConnectionDevice("Dima", "3458999"), new ConnectionDevice("Rita", "3458995"), false, 114, new DateTime(2018, 11, 10, 19, 17, 13));

List<CallDescriptor> con = new List<CallDescriptor>();

con.Add(dsr1);

con.Add(dsr2);

con.Add(dsr3);

return View(con);

}

} }

Рисунок 2 – Програмний код серверної частини програми

Дані прив’язуються до серверних елементів за допомогою спеціальних елементів управління. У нашому випадку це DataList(список). Елемент управління має властивість DataSource, що відповідає за прив’язку до даних. Тип цієї властивості – object, тобто він може бути будь-яким, але необхідно реалізувати інтерфейс IEnumerable. Значенням цієї властивості буде значення колекції списку «Call Descriptor».

Джерелом даних також можуть бути XML-файли, масиви, таблиці з баз даних. Викликом методу DataBind() дані безпосередньо прив’язуються до серверного елемента управління. Метод Page.DataBind() викликає прив’язку до усіх елементів на сторінці.

У нашому випадку для відображення інформації про робітника використовуємо DataList (список). Для прив’язки інформації про сутність, що представлена списком con, на боці серверу необхідно викликати код на рисунку 2.

View сутності «Дзвінок» буде виглядати наступним чином:

@model List<CallDescriptor>

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<title>Telephone Station</title>

<link href="~/css/site.css" rel="stylesheet" />

</head>

<body>

<h1>Call Descriptor</h1>

<table border="1" cellpadding="10">

<tr>

<th>Caller</th>

<th>Called</th>

<th>Activity</th>

<th>Duration</th>

<th>Start Call</th>

<th>End Call</th>

</tr>

@foreach (CallDescriptor dsr in Model)

{

<tr>

<td>@dsr.Caller.Phone</td>

<td>@dsr.Called.Phone</td>

<td>@dsr.IsActive</td>

<td>@dsr.Duration</td>

<td>@dsr.Start</td>

<td>@dsr.Start.AddSeconds(dsr.Duration)</td>

</tr>

}

</table>

</body>

</html>

Те, як дана сторінка буде виглядати зі стороні клієнта можна побачити на рисунку 3.

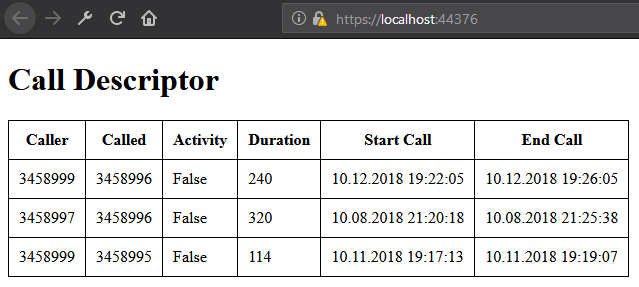


Рисунок 3 – Вигляд розробленого застосунку на боці клієнта

**Висновки:**

Виконуючи лабораторну роботу було проведено ознайомлення з прикладом додатку, що був представлений викладачем. Досліджено питання, що стосуються того: як данні представлено на боці серверу, яким чином дані представлено на боці клієнта, яким чином клієнтське уявлення зв’язується зі серверним, як здійснюється перевірка даних, що вводяться; як використовується та працює компонент DataList.