Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**Про виконання практичної роботи № 6**

**з дисципліни «Програмне забезпечення мережевих технологій»**

**на тему**

**«Створення веб сторінок засобами ASP.NET.**

**Робота з базами даних за допомогою стандартних бібліотек C#»**

**Лектор:**

професор кафедри ПЗ

Мельник Р. А.

**Виконав:**

студент групи ПІ-31

Губ’як Р. І.

**Прийняв:**

старший викладач кафедри ПЗ

Гасько Р. Т.

«\_\_\_» \_\_\_\_\_\_\_\_ 2017 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2017

**ТЕМА РОБОТИ**

Створення веб сторінок засобами ASP.NET. Робота з базами даних за допомогою стандартних бібліотек C#

**МЕТА РОБОТИ**

Ознайомитись з класами та методами середовища ASP.NET. Освоїти елементи мови C# для роботи з базами даних

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

Розробка ColorInteraction.

1. Реалізувати логування у базу даних.
2. Реалізувати візуалізацію колірної взаємодії через веб-браузер.

**ТЕКСТ ПРОГРАМИ**

//Робота з базою даних

public class Database

{

private readonly string ConnectionString;

public Database()

{

ConnectionString = "Server=RZR2PC\\RZR2SQL;Database=DbForDotnet;User Id = sa; password=Роман з Рясне-2;";

}

public async void Log(ColorMessage colorMessage)

{

SqlConnection sqlConnection = new SqlConnection(ConnectionString);

string color = JsonConvert.SerializeObject(colorMessage.Color);

string script = $"INSERT INTO Logs VALUES ('{colorMessage.MachineName}', '{color}', GETDATE());";

SqlCommand sqlCommand = new SqlCommand(script, sqlConnection);

sqlConnection.Open();

await sqlCommand.ExecuteNonQueryAsync();

sqlConnection.Close();

}

public async Task<List<ColorMessage>> GetColorMessages(DateTime lastUpdate)

{

SqlConnection sqlConnection = new SqlConnection(ConnectionString);

string script = @"SELECT MachineName, Color FROM Logs

WHERE Id IN (

SELECT MAX(Id) FROM Logs

WHERE LastUpdate > @lastUpdate

GROUP BY MachineName);";

SqlCommand sqlCommand = new SqlCommand(script, sqlConnection);

sqlCommand.Parameters.Add(new SqlParameter

{

ParameterName = "@lastUpdate",

SqlDbType = SqlDbType.DateTime,

Value = lastUpdate

});

sqlConnection.Open();

SqlDataReader reader = await sqlCommand.ExecuteReaderAsync();

List<ColorMessage> result = new List<ColorMessage>();

while (reader.Read())

{

result.Add(new ColorMessage

{

Color = JsonConvert.DeserializeObject<Color>(reader.GetString(reader.GetOrdinal("Color"))),

MachineName = reader.GetString(reader.GetOrdinal("MachineName"))

});

}

sqlConnection.Close();

return result;

}

}

-----------------------------------------------------------------------------View веб-сторінки

@{

ViewBag.Title = "ColorInteraction";

}

<h2>@ViewBag.Title.</h2>

<h3>@ViewBag.Message</h3>

<div id="color" class="col-xs-6" style="height: 50vh;"></div>

<table id="actors">

<tr>

<th>Machine</th>

<th>Color</th>

</tr>

</table>

<script type="text/javascript">

@{DateTime now = DateTime.Now;}

setInterval(function () {

$.ajax({

url: "@Url.HttpRouteUrl("DefaultApi", new

{

controller = "Color",

action = "CurrentColor",

lastUpdate = now

})",

success: function (result) {

$("#color").css("background-color", `rgb(${result.r}, ${result.g}, ${result.b})`);

}

});

$.ajax({

url: "@Url.HttpRouteUrl("DefaultApi", new

{

controller = "Color",

action = "Actors",

lastUpdate = now

})",

success: function (response) {

$("#actors").children().empty();

$("#actors>tbody").append(`<tr>

<th>Machine</th>

<th>Color</th>

</tr>`);

for (var i = 0; i < response.length; i++) {

$("#actors>tbody").append(`<tr>

<td>${response[i].machineName}</td>

<td style="background-color: rgb(${response[i].color.r}, ${response[i].color.g}, ${response[i].color.b}); width: 50vh;"></td>

</tr>`);

}

}

});

}, 3000);

</script>

//Контролер веб-сторінки

public class ColorController : ApiController

{

[HttpGet]

public async Task<Color> CurrentColor(DateTime lastUpdate)

{

List<ColorMessage> colorMessages = await new Database.Database()

.GetColorMessages(lastUpdate);

if (colorMessages.Count == 0)

{

return new Color();

}

byte r = (byte) colorMessages.Average(item => item.Color.R);

byte g = (byte) colorMessages.Average(item => item.Color.G);

byte b = (byte) colorMessages.Average(item => item.Color.B);

return new Color

{

R = r,

B = b,

G = g

};

}

[HttpGet]

public async Task<IEnumerable<ColorMessageModel>> Actors(DateTime lastUpdate)

{

return (await new Database.Database()

.GetColorMessages(lastUpdate))

.Select(item => new ColorMessageModel

{

MachineName = item.MachineName,

Color = new Color

{

R = item.Color.R,

G = item.Color.G,

B = item.Color.B

}

});

}

}

**ПРОТОКОЛ ВИКОНАННЯ ПРОГРАМИ**

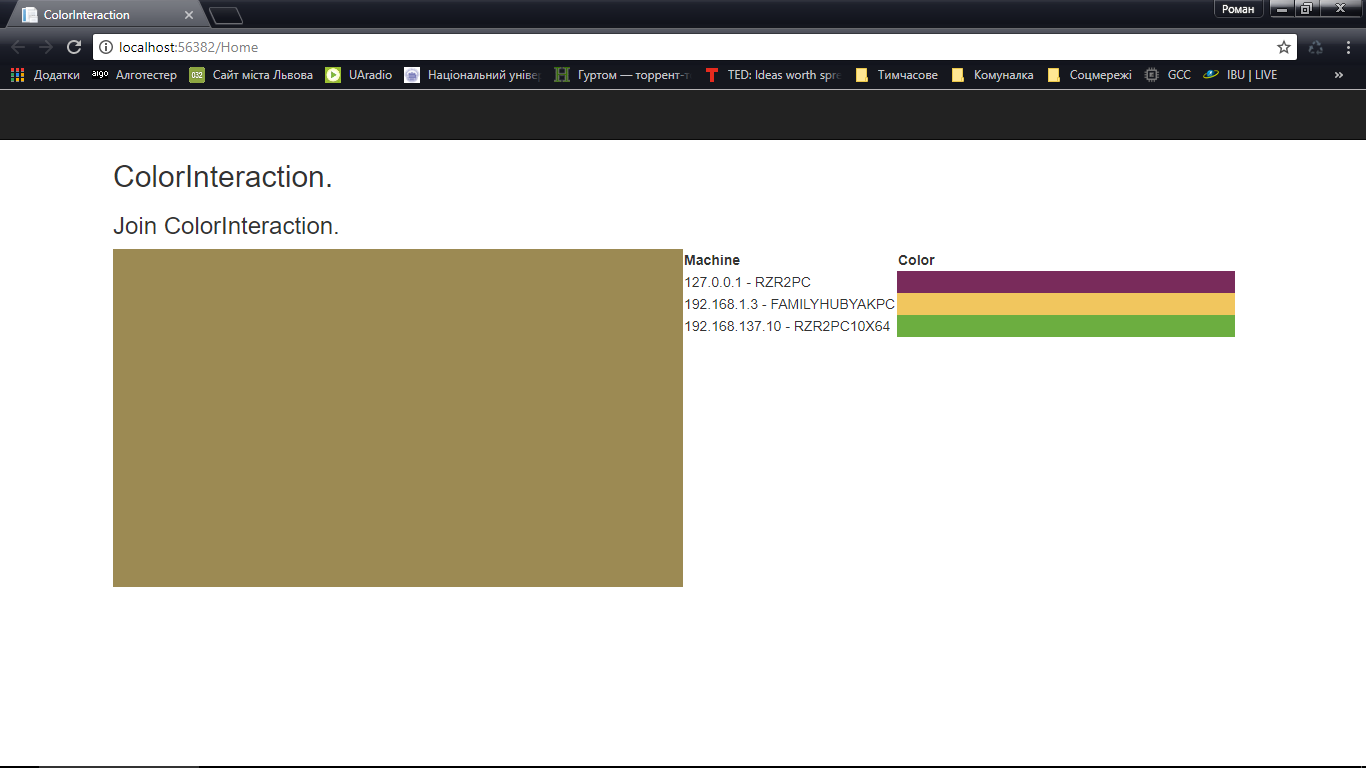


Рис. 1. Результат колірної взаємодії у вікні веб-браузера

**ВИСНОВОК**

Під час виконання цієї лабораторної роботи я ознайомився з класами та методами середовища ASP.NET. Освоїв елементи мови C# для роботи з базами даних. Реалізував динамічну веб-сторінку та частину системи ColorInteraction, яка відповідає за роботу з базою даних, згідно з індивідуальним завданням.