МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з лабораторної роботи №4

з дисципліни «Стек технологій .NET»

Виконав:

Студент групи КН-36а

Рубан Ю. Д.

Перевірила:

Доц. Літвінова Ю. С.

Харків – 2018

**Тема:** Початкове знайомство з простими WEB застосунками

**Ціль:** Вивчити базові принципи побудови та функціональності простіших WEB застосунків на платформі ASP.NET.

**Завдання:** Розробити веб-застосунок «Калькулятор», який реалізує основні математичні операції.

**Хід виконання роботи:**

Було розроблено веб-застосунок «Калькулятор», який реалізує основні математичні операції. Робота калькулятора показана на рисунку 1.

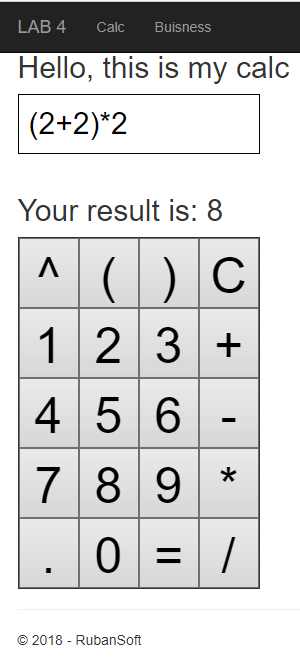


Рисунок 1 – Робота калькулятора

У лістингу 1 показаний код сторінки cshtml.

Лістинг 1. Код сторінки з калькулятором

@model double?

@{

ViewBag.Title = "Calc";

}

<script src="~/Scripts/jquery-1.10.2.min.js"></script>

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

<h2>Hello, this is my calc</h2>

@Html.ValidationSummary(false)

@using (Html.BeginForm("Index","Home",FormMethod.Post,new {@id = "calc-form" }))

{

@Html.TextBox("expression","",new {@class = "inp-expr" });

}

@if (Model != null)

{

<h2>Your result is: @Model</h2>

}

<div class="number-block">

<div class="numbers">

<button class="operation">^</button>

<button class="operation">(</button>

<button class="operation">)</button>

<button class="operation">C</button>

<button class="number number1">1</button>

<button class="number number2">2</button>

<button class="number number3">3</button>

<button class="operation">+</button>

<button class="number number4">4</button>

<button class="number number5">5</button>

<button class="number number6">6</button>

<button class="operation">-</button>

<button class="number number7">7</button>

<button class="number number8">8</button>

<button class="number number9">9</button>

<button class="operation">\*</button>

<button class="number dot">.</button>

<button class="number number0">0</button>

<button class="number eq">=</button>

<button class="operation">/</button>

</div>

</div>

<script>

var redirect = function() {

var lk = document.location.origin.toString() + "@Url.RouteUrl(new { controller = "Home", action = "Index" })";

document.location.href = lk.toString();

}

var nmb = document.getElementsByClassName("numbers")[0];

var btn = nmb.getElementsByTagName("button");

var inp = document.getElementById("expression");

jQuery("#expression").on("input", function () {

this.setAttribute("value", $("#expression").val());

})

for (var i = btn.length - 1; i >= 0; i--) {

let buf = btn[i];

if (buf.innerHTML != "C") {

btn[i].addEventListener("click", function () {

if (this.innerHTML == "C") {

inp.setAttribute("value", "");

this.addEventListener("click", redirect);

}

else if (this.innerHTML == "=") {

document.getElementById("calc-form").submit();

}

else {

let val = $("#expression").val();

val += (this.innerHTML);

$("#expression").val(val);

}

});

}

else {

buf.addEventListener("click", redirect);

}

}

</script>

У лістингу 2 показаний код контролера, який відповідає за відображення сторінки.

Лістинг 2. Код контролера

using lab4Calc.Models;

using System.Web.Mvc;

namespace lab4Calc.Controllers

{

public class HomeController : Controller

{

private IExpressionParser parser;

public HomeController(IExpressionParser parser)

{

this.parser = parser;

}

public ActionResult Index()

{

return View();

}

[HttpPost]

public ActionResult Index(string expression)

{

parser.Expression = expression;

if (!parser.CheckSyntax())

{

ModelState.AddModelError("syntax", "Выражение введено не правильно");

}

double func=0;

try

{

func = parser.Parse()();

}

catch

{

ModelState.AddModelError("", "Ошибка вычисления");

}

if (!ModelState.IsValid)

{

return View();

}

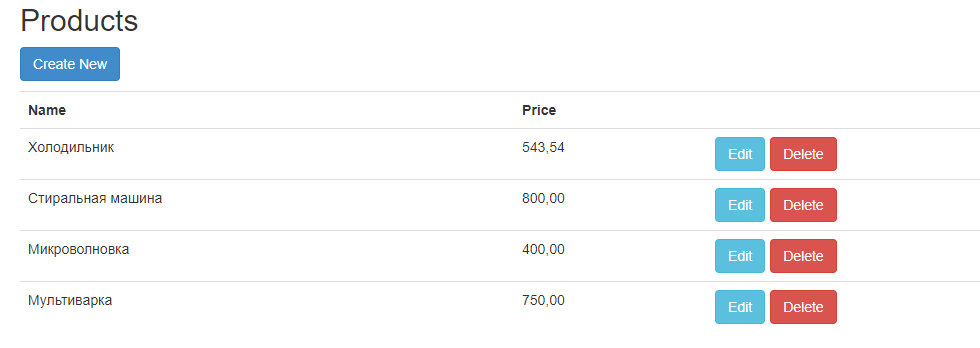
return View(func);

}

}

}

Також для заданої предметної області було реалізовано базовий функціонал CRUD. Робота функціоналу показана на рисунках 2 – 4.

Рисунок 2 – Таблиця з продуктами

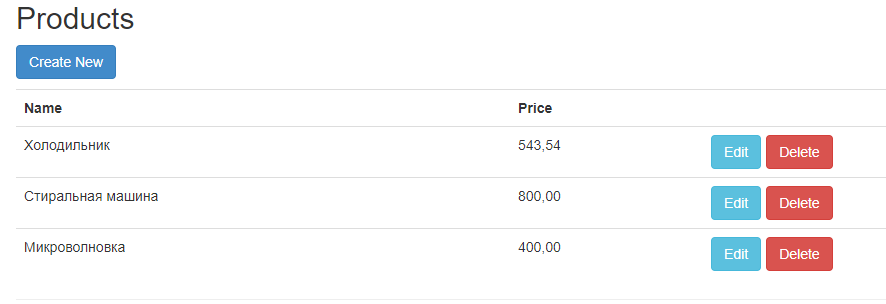
Після натискання кнопки Delete обраний товар видаляється. Рисунок 3.

Рисунок 3 – Таблиця товарів після видалення одного з них

Після натискання кнопки Edit або Create New програма переходить на сторінку редагування товару це показано на рисунку 4.

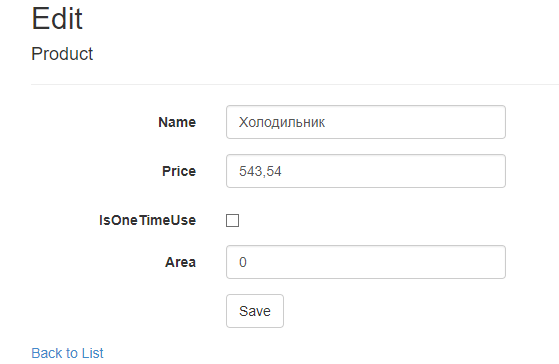


Рисунок 4 – редагування товару.

У лістингу 3 показаний контролер який відповідає за CRUD операції з товаром.

using lab4Calc.Models.Buisness;

using lab4Calc.Models.Data;

using System.Web.Mvc;

namespace lab4Calc.Controllers

{

public class BuisnessController : Controller

{

IProductRepository repository;

public BuisnessController(IProductRepository repository)

{

this.repository = repository;

}

public ActionResult Index()

{

return View(repository.Products);

}

[HttpPost]

public ActionResult Delete(int Id)

{

repository.Remove(repository.GetById(Id));

return RedirectToAction("Index");

}

public ActionResult Edit(int? Id)

{

if(Id == null)

{

return View(new Product());

}

else

{

return View(repository.GetById(Id.Value));

}

}

[HttpPost]

public ActionResult Edit(Product product)

{

repository.Update(product);

return RedirectToAction("Index");

}

}

}

**Висновки:** У даній лабораторній роботі було вивчено базові принципи побудови та функціональності простих веб застосунків на платформі ASP.NET