МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з лабораторної роботи №5

з дисципліни «Стек технологій .NET»

Виконав:

Студент групи КН-36а

Рубан Ю. Д.

Перевірила:

Доц. Літвінова Ю. С.

Харків – 2018

**Тема:** Робота з даними в ASP.NET

**Ціль:** Познайомитися з рядом засобів ASP.NET для представлення та відображення даних.

**Варіант індивідуального завдання:** Виробниче підприємство по виробництву побутової техніки.

**Опис механізму роботи з даними розробленого застосунку:**

Коли клієнт заходить на сайт він може обрати товар та добавити його до кошика. На сервері за цей процес відповідає контролер CartController який зберігає товари у кошик, та формує замовлення для подальшої його обробки. Лістинг контролера показаний на лістингу 1.

Лістинг 1. Код контролера CartController

using lab4Calc.Models;

using lab4Calc.Models.Buisness;

using System.Collections.Generic;

using System.Web.Mvc;

using static lab4Calc.Models.Buisness.Order;

namespace lab4Calc.Controllers

{

public class CheckoutViewModel

{

public OrderInfo OrderInfo { get; set; }

public string ReturnUrl { get; set; }

}

public class CartController : Controller

{

private CustomerFacade facade;

public CartController(CustomerFacade facade)

{

this.facade = facade;

}

public ActionResult CartPreview(Cart cart)

{

return PartialView(cart);

}

public ActionResult Details(Cart cart, string returnUrl)

{

if(returnUrl == null)

{

returnUrl = Url.Action("Index", "Buisness");

}

return View(new CartViewModel() {Cart = cart,ReturnUrl = returnUrl });

}

public ActionResult Remove(Cart cart,int id, string returnUrl)

{

cart.RemoveLine(id);

return RedirectToAction("Details", new { returnUrl});

}

public ActionResult Checkout(string returnUrl)

{

return View(new CheckoutViewModel() {OrderInfo= new OrderInfo(),ReturnUrl = returnUrl });

}

[HttpPost]

public ActionResult Checkout(Cart cart, CheckoutViewModel checkout)

{

if (ModelState.IsValid)

{

Stack<Product> temp = new Stack<Product>();

foreach(var p in cart.Lines)

{

for (int i = 0; i < p.Quntity; i++)

{

temp.Push(p.Product);

}

}

var pay = facade.MakeOrder(temp, checkout.OrderInfo);

TempData["Pay"] = pay;

return RedirectToAction("Pay",new {returnUrl=checkout.ReturnUrl });

}

return View(checkout);

}

public ActionResult Pay(string returnUrl)

{

ViewBag.returnUrl = returnUrl;

return View((PayBuilder)TempData.Peek("Pay"));

}

[HttpPost]

public ActionResult Pay(Cart cart, decimal money,string returnUrl, MadedOrders madedOrders)

{

var pay = (PayBuilder)TempData["Pay"];

if (ModelState.IsValid)

{

try

{

Order order = pay.Pay(money);

facade.MakeProducts(order);

madedOrders.Orders.Add(order);

cart.Clear();

return RedirectPermanent(returnUrl);

}

catch

{

ModelState.AddModelError("", "При оплате возникла непредвиденная ошибка, возможно вы заплатили сумму меньше указаной");

return View(pay);

}

}

else

{

return View(pay);

}

}

}

}

Далі клієнт підтверджує своє замовлення і на цьому цикл роботи замовлення товарів завершується. Лістинг роботи контролеру, який відповідає за обробку замовлень показаний у лістингу 2.

Лістинг 2. OrderController

using lab4Calc.Models;

using lab4Calc.Models.Buisness;

using lab4Calc.Models.Data;

using System.Web.Mvc;

namespace lab4Calc.Controllers

{

public class OrderController : Controller

{

private CustomerFacade facade;

private IOrderRepository repository;

public OrderController(IOrderRepository repository,CustomerFacade facade)

{

this.facade = facade;

this.repository = repository;

}

public ActionResult Index(MadedOrders orders)

{

return View(orders.Orders);

}

public ActionResult OrderPreview(MadedOrders orders)

{

return PartialView(orders);

}

[HttpPost]

public ActionResult Accept(int index, MadedOrders madedOrders)

{

Order order = madedOrders.Orders[index];

repository.Update(order);

madedOrders.Orders.Remove(order);

TempData["DP"] = facade.UnpackDeliveredProducts(facade.DeliverProducts(order));

return RedirectToAction("Index","Buisness");

}

}

}

Конструктор цього класу отримує об’єкт інтерфейсу IOrderRepository, який відповідає за роботу з заявками. Контролеру не важливо як конкретно реалізований цей інтерфейс і тому, логіку зберігання заявок завжди можна замінити.

Для відображення даних технологія ASP.NET MVC використовує Razor view engine, який отримує об'єкт моделі з контролеру, та дозволяє використовувати C# код всередені HTML розмітки. Файл який містить HTML та C# код називається уявленням та має розширення .cshtml. Лістинг одного з таких файлів показаний у лістингу 3.

Лістинг 3. Приклад cshtml файлу

@model lab4Calc.Models.CartViewModel

@{

ViewBag.Title = "Details";

}

<h2>Details</h2>

<table class="table table-bordered table-hover">

<tbody>

<tr>

<th>Количество</th>

<th>Название</th>

<th>Цена</th>

<th colspan="2">Всего</th>

</tr>

@foreach (var p in Model.Cart.Lines)

{

<tr>

<td>@p.Quntity</td>

<td>@p.Product.Name</td>

<td>@p.Product.Price</td>

<td>@p.LinePrice</td>

<td>

@using (Html.BeginForm("Remove", "Cart", new { id = p.Product.Id }, FormMethod.Post))

{

@Html.HiddenFor(x=>x.ReturnUrl)

<button class="btn btn-danger" type="submit">Удалить</button>

}

</td>

</tr>

}

</tbody>

@if (Model.Cart.TotalQuantity > 0)

{

<tfoot>

<tr>

<td colspan="1">

Итого

</td>

<td>

Товаров: @Model.Cart.TotalQuantity

</td>

<td>

Всего к оплате: @Model.Cart.TotalPrice

</td>

<td>

@using (Html.BeginForm("Checkout", "Cart", FormMethod.Get))

{

@Html.Hidden("returnUrl",Model.ReturnUrl)

<button class="btn btn-primary" type="submit">Оформить заказ</button>

}

</td>

<td>

<a href="@Model.ReturnUrl" class="btn btn-default">Вернуться назад</a>

</td>

</tr>

</tfoot>

}

else

{

<tfoot>

<tr>

<td colspan="4">

Корзина пуста

</td>

<td>

<a href="@Model.ReturnUrl" class="btn btn-default">Вернуться назад</a>

</td>

</tr>

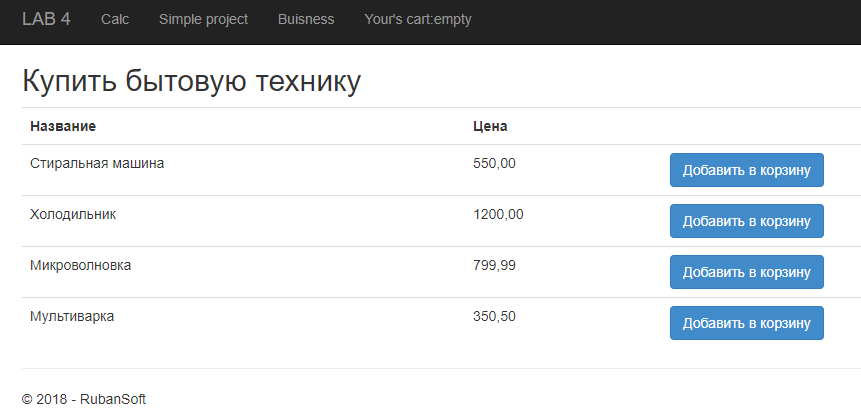
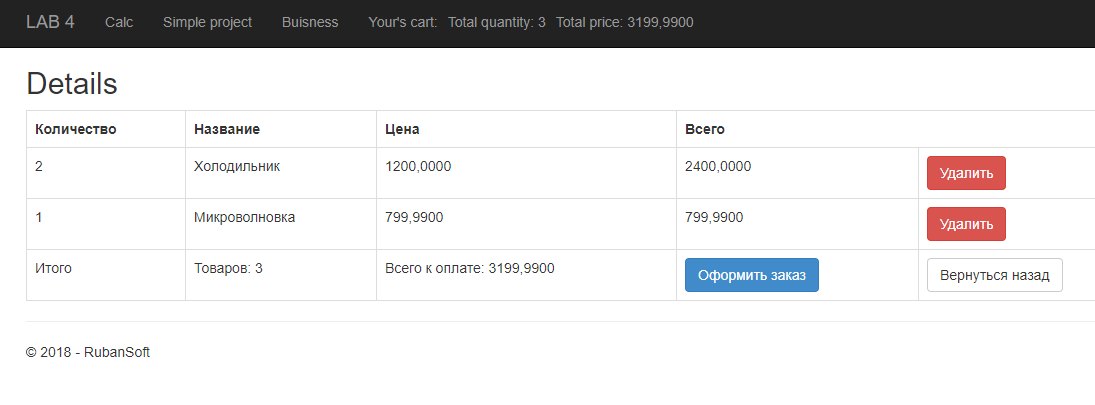
</tfoot>

}

</table>

Даний файл відповідає за відображення товарів у корзині.

Розроблений веб застосунок показаний на рисунках 1 – 5.

Рисунок 1 – Вибір товарів

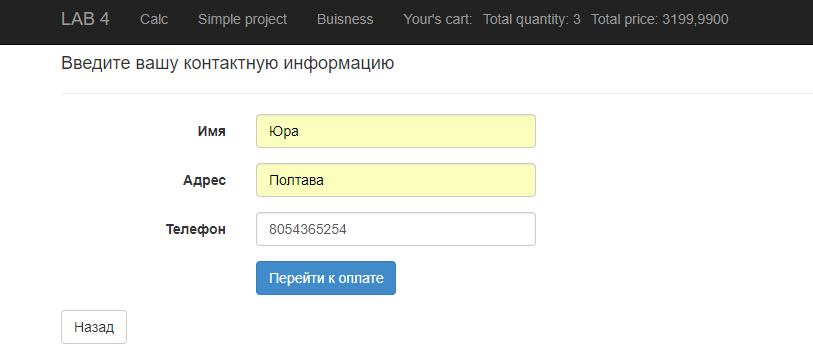
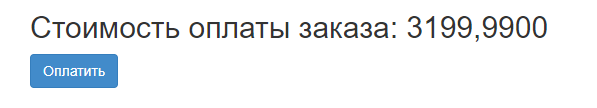
Рисунок 2 – Кошик клієнта

Рисунок 3 – Інформація для доставки.



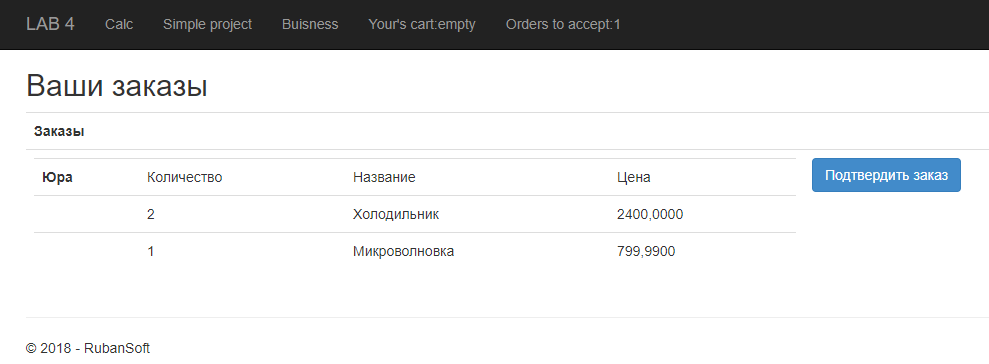
Рисунок 4 – Оплата товарів

Рисунок 5 – Підтвердженя замовлення

**Висновки:**

У даній лабораторній роботі було ознайомлено з рядом засобів ASP.NET для представлення та відображення даних.