**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №7

по дисциплине: **«РПБДИС»**

на тему: **Создание *ASP*.*NET* приложения в заданной предметной области средствами *MS* *Visual* *Studio* на базе *Web*-форм**

Выполнил: студент гр. ИТП-31

Сетко А.И.

Принял: доцент

Асенчик О.Д.

Гомель 2019

**Цель**: ознакомиться с возможностями среды разработки *Microsoft* *Visual* *Studio* по созданию *Web*-приложений баз данных, в том числе с использование специализированных элементов управления — поставщиков и потребителей данных. Научиться использовать средства визуальной среды разработки для создания *web*-приложений баз данных на основе технологии *ADO*.*NET* и *Web*-форм.

**Задание:** используя среду *Microsoft* *Visual* *Studio*, спроектировать и создать *Web*-приложение на основе набора *Web*-форм.

Необходимо создать систему форм для последующей работы с данными согласно вашему варианту, а также систему навигации по приложению.

*Web*-приложение должно удовлетворять следующим требованиям:

1. Вход в приложение возможен только для аутентифицированных пользователей.

2. Все формы *web*-приложения должны быть построены на базе шаблона, задаваемого одной или несколькими *Master* *Page* страницами.

3. Система навигация должна присутствовать и обеспечивать оптимальный путь перехода между двумя произвольно выбранными формами в соответствии с логикой приложения.

4. Система навигации должна включать карту сайта, построенную на основе данных специализированного файла *web*.*sitemap*.

5. Для создания системы навигации использовать элементы управления *TreeView*, *Menu* и *SiteMapPath*.

6. *Web*-формы должны использовать преимущественно форму табличного представление информации из базы данных.

7. Осуществлять ввод, редактирование, добавление и просмотра данных реляционной базы данных согласно варианту в удобном для пользователя виде.

8. *Web*-формы должны содержать текстовые поля, поля со списком, табличные элементы управления (*GridView*, *ListView*), командные кнопки и другие элементы.

9. При разработке *Web*-форм предусмотреть, где необходимо, проверку вводимых значений перед их отправкой на сервер.

**Ход работы**

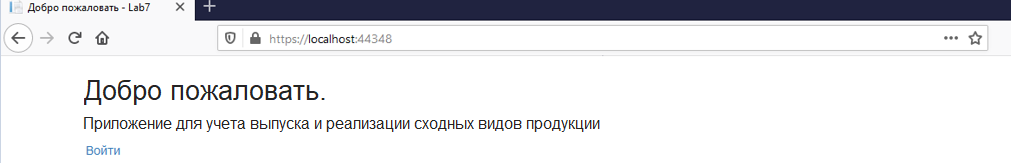


Рисунок 1 – Копия экрана браузера главной страницы сайта для неавторизованного пользователя

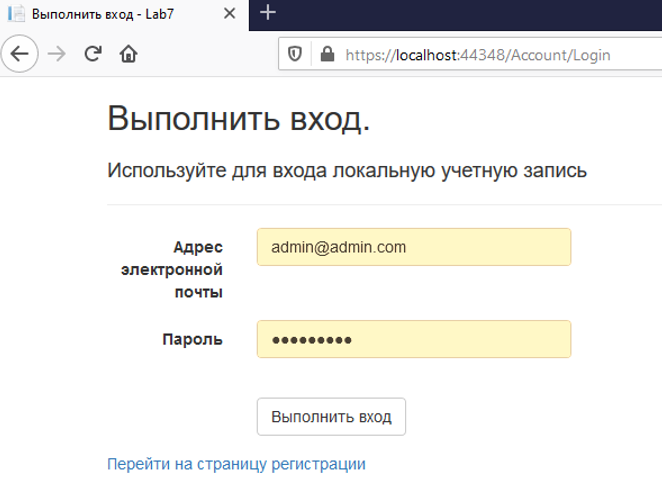


Рисунок 2 – Копия экрана браузера страницы для входа в аккаунт

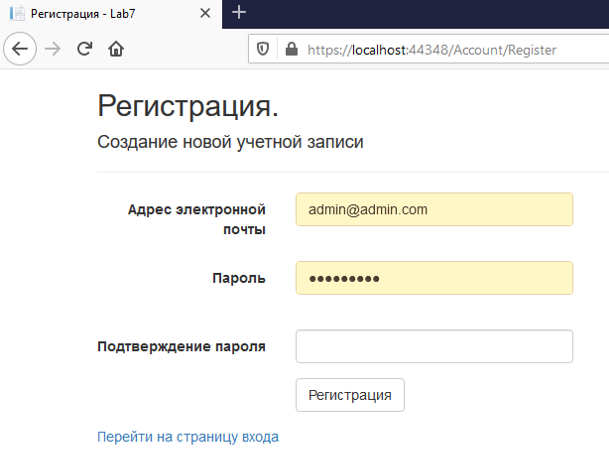


Рисунок 3 – Копия экрана браузера страницы регистрации аккаунта

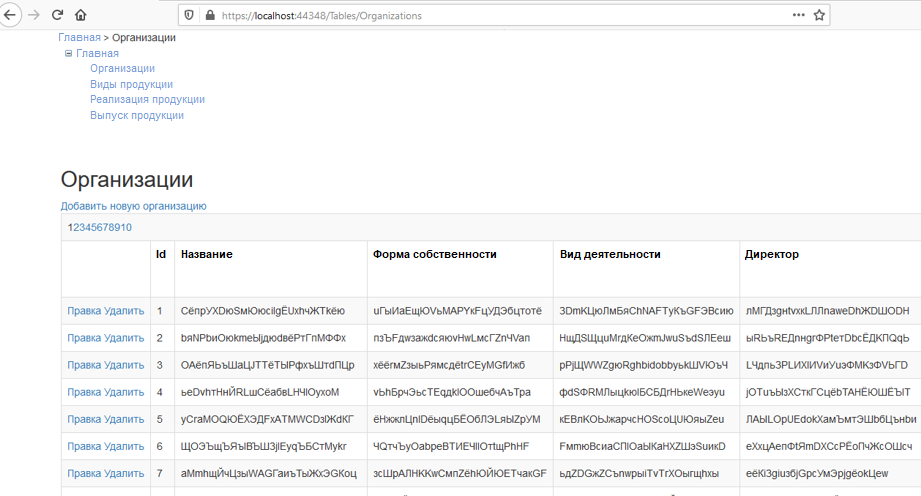


Рисунок 4 – Копия экрана браузера страницы со списком организаций

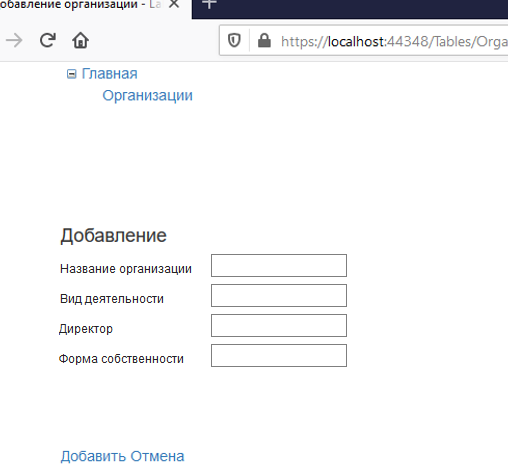


Рисунок 5 – Копия экрана браузера страницы добавления организации

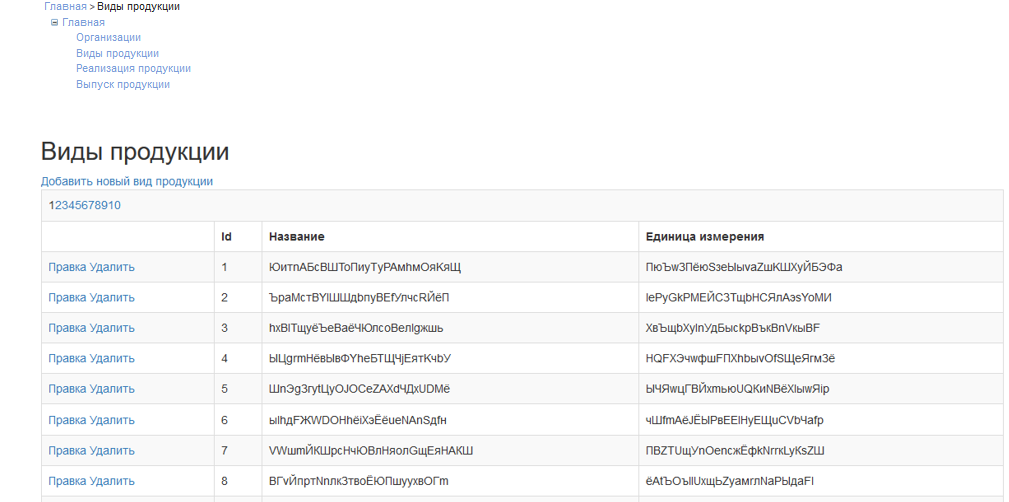


Рисунок 6 – Копия экрана браузера страницы со списком видов продукции

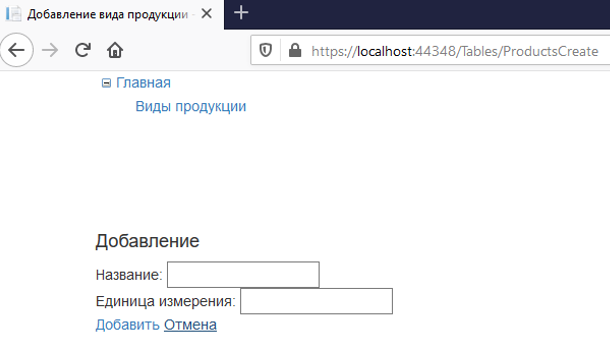


Рисунок 7 – Копия экрана браузера страницы добавления вида продукции

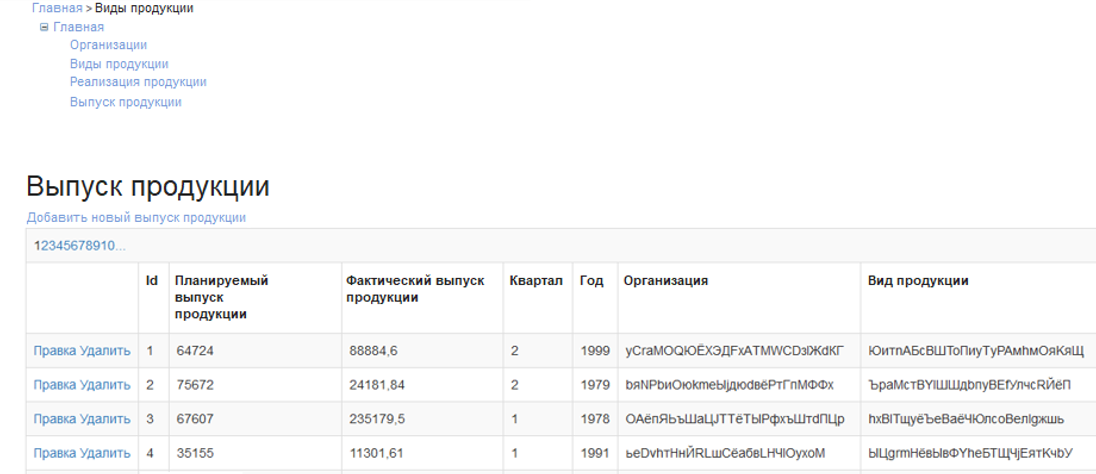


Рисунок 8 – Копия экрана браузера страницы со списком выпуска продукции

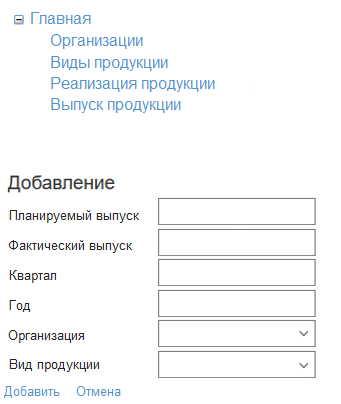


Рисунок 9 – Копия экрана браузера страницы добавления выпуска продукции

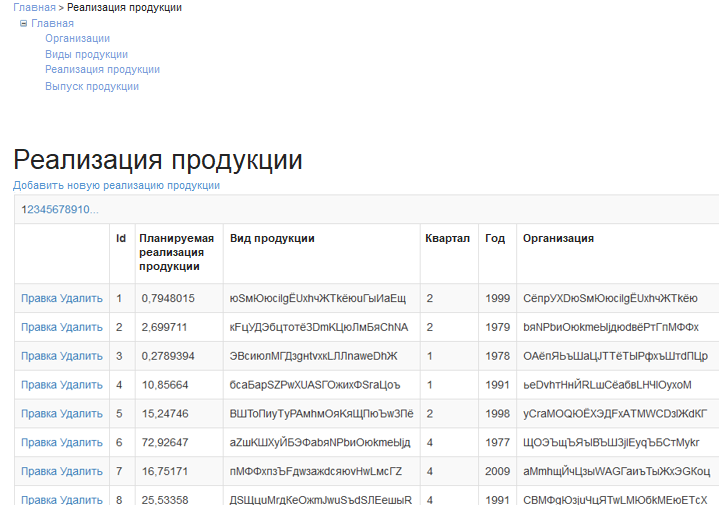


Рисунок 10 – Копия экрана браузера страницы со списком реализации продукции

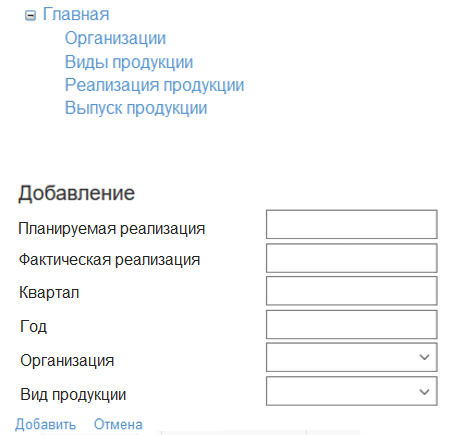


Рисунок 11 – Копия экрана браузера страницы добавления реализации продукции

**Вывод**: в результате выполнения лабораторной работы были ознакомлены с возможностями среды разработки *Microsoft* *Visual* *Studio* по созданию *Web*-приложений баз данных, в том числе с использование специализированных элементов управления — поставщиков и потребителей данных. Также были изучены средства визуальной среды разработки для создания *web*-приложений баз данных на основе технологии *ADO*.*NET* и *Web*-форм.

**Приложение А**

**Исходный код программы**

**Site.Master:**

<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs" Inherits="Lab7.SiteMaster" %>

<!DOCTYPE html>

<html lang="ru">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title><%: Page.Title %> - Lab7</title>

<asp:PlaceHolder runat="server">

<%: Scripts.Render("~/bundles/modernizr") %>

</asp:PlaceHolder>

<webopt:bundlereference runat="server" path="~/Content/css" />

<link href="~/favicon.ico" rel="shortcut icon" type="image/x-icon" />

</head>

<body>

<form runat="server">

<asp:ScriptManager runat="server">

<Scripts>

<%--To learn more about bundling scripts in ScriptManager see https://go.microsoft.com/fwlink/?LinkID=301884 --%>

<%--Framework Scripts--%>

<asp:ScriptReference Name="MsAjaxBundle" />

<asp:ScriptReference Name="jquery" />

<asp:ScriptReference Name="bootstrap" />

<asp:ScriptReference Name="WebForms.js" Assembly="System.Web" Path="~/Scripts/WebForms/WebForms.js" />

<asp:ScriptReference Name="WebUIValidation.js" Assembly="System.Web" Path="~/Scripts/WebForms/WebUIValidation.js" />

<asp:ScriptReference Name="MenuStandards.js" Assembly="System.Web" Path="~/Scripts/WebForms/MenuStandards.js" />

<asp:ScriptReference Name="GridView.js" Assembly="System.Web" Path="~/Scripts/WebForms/GridView.js" />

<asp:ScriptReference Name="DetailsView.js" Assembly="System.Web" Path="~/Scripts/WebForms/DetailsView.js" />

<asp:ScriptReference Name="TreeView.js" Assembly="System.Web" Path="~/Scripts/WebForms/TreeView.js" />

<asp:ScriptReference Name="WebParts.js" Assembly="System.Web" Path="~/Scripts/WebForms/WebParts.js" />

<asp:ScriptReference Name="Focus.js" Assembly="System.Web" Path="~/Scripts/WebForms/Focus.js" />

<asp:ScriptReference Name="WebFormsBundle" />

<%--Site Scripts--%>

</Scripts>

</asp:ScriptManager>

<div class="container body-content">

<div id="sitemap" runat="server" visible="false">

<asp:SiteMapPath ID="SiteMapPath1" runat="server" />

<asp:TreeView ID="TreeView1" runat="server" DataSourceID="SiteMapDataSource1" />

<asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server" />

<br />

<br />

</div>

<asp:ContentPlaceHolder ID="MainContent" runat="server">

</asp:ContentPlaceHolder>

<hr />

<footer>

<p>&copy; <%: DateTime.Now.Year %> - Lab7 – Setko A.I.</p>

</footer>

</div>

</form>

</body>

</html>

**Site.Master.cs:**

using System;

using System.Web;

using System.Web.Security;

using System.Web.UI;

namespace Lab7

{

public partial class SiteMaster : MasterPage

{

private const string AntiXsrfTokenKey = "\_\_AntiXsrfToken";

private const string AntiXsrfUserNameKey = "\_\_AntiXsrfUserName";

private string \_antiXsrfTokenValue;

protected void Page\_Init(object sender, EventArgs e)

{

// Код ниже защищает от XSRF-атак

var requestCookie = Request.Cookies[AntiXsrfTokenKey];

if (requestCookie != null && Guid.TryParse(requestCookie.Value, out Guid requestCookieGuidValue))

{

// Использование маркера Anti-XSRF из файла cookie

\_antiXsrfTokenValue = requestCookie.Value;

Page.ViewStateUserKey = \_antiXsrfTokenValue;

}

else

{

// Создание нового маркера Anti-XSRF и его сохранение в файле cookie

\_antiXsrfTokenValue = Guid.NewGuid().ToString("N");

Page.ViewStateUserKey = \_antiXsrfTokenValue;

var responseCookie = new HttpCookie(AntiXsrfTokenKey)

{

HttpOnly = true,

Value = \_antiXsrfTokenValue

};

if (FormsAuthentication.RequireSSL && Request.IsSecureConnection)

{

responseCookie.Secure = true;

}

Response.Cookies.Set(responseCookie);

}

Page.PreLoad += Master\_Page\_PreLoad;

}

protected void Master\_Page\_PreLoad(object sender, EventArgs e)

{

if (!IsPostBack)

{

// Задание маркера Anti-XSRF

ViewState[AntiXsrfTokenKey] = Page.ViewStateUserKey;

ViewState[AntiXsrfUserNameKey] = Context.User.Identity.Name ?? String.Empty;

}

else

{

// Проверка маркера Anti-XSRF

if ((string)ViewState[AntiXsrfTokenKey] != \_antiXsrfTokenValue

|| (string)ViewState[AntiXsrfUserNameKey] != (Context.User.Identity.Name ?? String.Empty))

{

throw new InvalidOperationException("Ошибка проверки маркера Anti-XSRF.");

}

}

}

protected void Page\_Load(object sender, EventArgs e)

{

sitemap.Visible = HttpContext.Current.User.Identity.IsAuthenticated;

}

}

}

**Default.aspx:**

<%@ Page Title="Добро пожаловать" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="Lab7.Default" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<h1><%: Title %>.</h1>

<h4>Приложение для учета и реализации сходных видов продукции</h4>

<div id="loginDiv" runat="server">

<a runat="server" href="~/Account/Login">Войти</a>

</div>

<div id="logoutDiv" runat="server">

<asp:LoginStatus runat="server" LogoutAction="Redirect" LogoutText="Выход" LogoutPageUrl="~/" OnLoggingOut="Unnamed\_LoggingOut" />

</div>

</asp:Content>

**Default.aspx.cs:**

using Microsoft.AspNet.Identity;

using System;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Lab7

{

public partial class Default : Page

{

protected void Page\_Load(object sender, EventArgs e)

{

loginDiv.Visible = !User.Identity.IsAuthenticated;

logoutDiv.Visible = User.Identity.IsAuthenticated;

}

protected void Unnamed\_LoggingOut(object sender, LoginCancelEventArgs e)

{

Context.GetOwinContext().Authentication.SignOut(DefaultAuthenticationTypes.ApplicationCookie);

}

}

}

**Login.aspx:**

<%@ Page Title="Выполнить вход" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Login.aspx.cs" Inherits="Lab7.Account.Login" Async="true" %>

<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">

<h2><%: Title %>.</h2>

<div class="row">

<div class="col-md-8">

<section id="loginForm">

<div class="form-horizontal">

<h4>Используйте для входа локальную учетную запись</h4>

<hr />

<asp:PlaceHolder runat="server" ID="ErrorMessage" Visible="false">

<p class="text-danger">

<asp:Literal runat="server" ID="FailureText" />

</p>

</asp:PlaceHolder>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Email" CssClass="col-md-2 control-label">Адрес электронной почты</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Email" CssClass="form-control" TextMode="Email" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Email"

CssClass="text-danger" ErrorMessage="Поле адреса электронной почты заполнять обязательно" />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Password" CssClass="col-md-2 control-label">Пароль</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Password" TextMode="Password" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Password" CssClass="text-danger" ErrorMessage="Поле пароля заполнять обязательно" />

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<asp:Button runat="server" OnClick="LogIn" Text="Выполнить вход" CssClass="btn btn-default" />

</div>

</div>

<div class="form-group">

<div class="col-md-4">

<a runat="server" href="~/Account/Register">Перейти на страницу регистрации</a>

</div>

</div>

</div>

</section>

</div>

</div>

</asp:Content>

**Login.aspx.cs:**

using System;

using System.Web;

using System.Web.UI;

using Microsoft.AspNet.Identity;

using Microsoft.AspNet.Identity.Owin;

using Owin;

using Lab7.Models;

namespace Lab7.Account

{

public partial class Login : Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void LogIn(object sender, EventArgs e)

{

if (IsValid)

{

// Проверка пароля пользователя

var signinManager = Context.GetOwinContext().GetUserManager<ApplicationSignInManager>();

// Сбои при входе не приводят к блокированию учетной записи

// Чтобы ошибки при вводе пароля инициировали блокирование, замените на shouldLockout: true

var result = signinManager.PasswordSignIn(Email.Text, Password.Text, false, shouldLockout: false);

switch (result)

{

case SignInStatus.Success:

IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);

break;

case SignInStatus.LockedOut:

Response.Redirect("/Account/Lockout");

break;

case SignInStatus.RequiresVerification:

Response.Redirect(String.Format("/Account/TwoFactorAuthenticationSignIn?ReturnUrl={0}&RememberMe={1}",

Request.QueryString["ReturnUrl"],

false),

true);

break;

case SignInStatus.Failure:

default:

FailureText.Text = "Неудачная попытка входа";

ErrorMessage.Visible = true;

break;

}

}

}

}

}

**Register.aspx:**

<%@ Page Title="Регистрация" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Register.aspx.cs" Inherits="Lab7.Account.Register" %>

<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">

<h2><%: Title %>.</h2>

<p class="text-danger">

<asp:Literal runat="server" ID="ErrorMessage" />

</p>

<div class="form-horizontal">

<h4>Создание новой учетной записи</h4>

<hr />

<asp:ValidationSummary runat="server" CssClass="text-danger" />

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Email" CssClass="col-md-2 control-label">Адрес электронной почты</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Email" CssClass="form-control" TextMode="Email" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Email"

CssClass="text-danger" ErrorMessage="Поле адреса электронной почты заполнять обязательно" />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="Password" CssClass="col-md-2 control-label">Пароль</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="Password" TextMode="Password" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="Password"

CssClass="text-danger" ErrorMessage="Поле пароля заполнять обязательно" />

</div>

</div>

<div class="form-group">

<asp:Label runat="server" AssociatedControlID="ConfirmPassword" CssClass="col-md-2 control-label">Подтверждение пароля</asp:Label>

<div class="col-md-10">

<asp:TextBox runat="server" ID="ConfirmPassword" TextMode="Password" CssClass="form-control" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="ConfirmPassword"

CssClass="text-danger" Display="Dynamic" ErrorMessage="Поле подтверждения пароля заполнять обязательно" />

<asp:CompareValidator runat="server" ControlToCompare="Password" ControlToValidate="ConfirmPassword"

CssClass="text-danger" Display="Dynamic" ErrorMessage="Пароль и его подтверждение не совпадают" />

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<asp:Button runat="server" OnClick="CreateUser\_Click" Text="Регистрация" CssClass="btn btn-default" />

</div>

</div>

<div class="form-group">

<div class="col-md-4">

<a runat="server" href="~/Account/Login">Перейти на страницу входа</a>

</div>

</div>

</div>

</asp:Content>

**Register.aspx.cs:**

using System;

using System.Linq;

using System.Web;

using System.Web.UI;

using Microsoft.AspNet.Identity;

using Microsoft.AspNet.Identity.Owin;

using Owin;

using Lab7.Models;

using Microsoft.AspNet.Identity.EntityFramework;

namespace Lab7.Account

{

public partial class Register : Page

{

protected void CreateUser\_Click(object sender, EventArgs e)

{

var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();

var signInManager = Context.GetOwinContext().Get<ApplicationSignInManager>();

var user = new ApplicationUser() { UserName = Email.Text, Email = Email.Text };

IdentityResult result = manager.Create(user, Password.Text);

if (result.Succeeded)

{

signInManager.SignIn( user, isPersistent: false, rememberBrowser: false);

IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);

if (!manager.IsInRole(manager.FindByEmail(user.Email).Id, "user"))

{

manager.AddToRole(manager.FindByEmail(user.Email).Id, "user");

}

}

else

{

ErrorMessage.Text = result.Errors.FirstOrDefault();

}

}

}

}

**Organizations.aspx:**

<%@ Page Title="Организации" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Organizations.aspx.cs" Inherits="Lab7.Organizations" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<section>

<div>

<hgroup>

<h2><%: Page.Title %></h2>

</hgroup>

<asp:HyperLink NavigateUrl="~/Tables/OrganizationsCreate" Text="Добавить новую организацию" runat="server" />

<asp:ValidationSummary ShowModelStateErrors="true" runat="server" />

<asp:GridView ID="organizationView" runat="server" AutoGenerateColumns="False" ShowFooter="True" GridLines="Vertical" CellPadding="4"

ItemType="Lab7.Models.Organization" SelectMethod="GetOrganizations" UpdateMethod="OrganizationView\_UpdateItem" DeleteMethod="OrganizationView\_DeleteItem" AutoGenerateEditButton="true" AutoGenerateDeleteButton="true"

CssClass="table table-striped table-bordered" AllowPaging="True">

<Columns>

<asp:BoundField DataField="Id" HeaderText="Id"/>

<asp:BoundField DataField="Name" HeaderText="Название"/>

<asp:BoundField DataField="FormOfOwnership" HeaderText="Форма собственности"/>

<asp:BoundField DataField="ActivityType" HeaderText="Вид деятельности" />

<asp:BoundField DataField="HeadName" HeaderText="Директор"/>

</Columns>

<PagerSettings Position="TopAndBottom" />

</asp:GridView>

</div>

</section>

</asp:Content>

**Organizations.aspx.cs:**

using Lab7.Models;

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Web.UI;

namespace Lab7

{

public partial class Organizations : Page

{

private readonly RationingDbContext \_db = new RationingDbContext();

protected void Page\_Load(object sender, EventArgs e)

{

}

public IOrderedQueryable<Organization> GetOrganizations()

{

IOrderedQueryable<Organization> query = \_db.Organizations.OrderBy(item => item.Id);

return query;

}

public void OrganizationView\_UpdateItem(int id)

{

Organization item = \_db.Organizations.Find(id);

if (item == null)

{

ModelState.AddModelError("", string.Format("Элемент с идентификатором {0} не был найден", id));

return;

}

TryUpdateModel(item);

if (ModelState.IsValid)

{

\_db.SaveChanges();

}

}

public void OrganizationView\_DeleteItem(int id)

{

var item = new Organization { Id = id };

\_db.Entry(item).State = EntityState.Deleted;

try

{

\_db.SaveChanges();

}

catch (DbUpdateConcurrencyException)

{

ModelState.AddModelError("", string.Format("Item with id {0} no longer exists in the database.", id));

}

}

}

}

**OrganizationsCreate.aspx:**

<%@ Page Title="Добавление организации" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="OrganizationsCreate.aspx.cs" Inherits="Lab7.Tables.OrganizationsCreate" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<h4>Добавление</h4>

<asp:ValidationSummary runat="server" ShowModelStateErrors="true" CssClass="error" />

<asp:FormView runat="server" ID="createOrganizationForm"

ItemType="Lab7.Models.Organization"

InsertMethod="CreateOrganizationForm\_InsertItem" DefaultMode="Insert"

RenderOuterTable="false" OnItemInserted="CreateOrganizationForm\_ItemInserted">

<InsertItemTemplate>

Название организации:

<asp:TextBox ID="NameTextBox" runat="server" Text='<%# Bind("Name") %>' />

<br />

Форма собственности:

<asp:TextBox ID="FormOfOwnershipTextBox" runat="server" Text='<%# Bind("FormOfOwnership") %>' />

<br />

Вид деятельности:

<asp:TextBox ID="ActivityTypeTextBox" runat="server" Text='<%# Bind("ActivityType") %>' />

<br />

ФИО директора:

<asp:TextBox ID="HeadNameTextBox" runat="server" Text='<%# Bind("NameOfDirector") %>' />

<br />

<asp:LinkButton runat="server" CausesValidation="True" CommandName="Insert" Text="Добавить" />

<asp:LinkButton runat="server" CausesValidation="False" CommandName="Cancel" Text="Отмена" OnClick="Unnamed\_Click" />

</InsertItemTemplate>

</asp:FormView>

</asp:Content>

**OrganizationsCreate.aspx.cs:**

using Lab7.Models;

using System;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Lab7.Tables

{

public partial class OrganizationsCreate : Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

public void CreateOrganizationForm\_InsertItem()

{

var item = new Organization();

TryUpdateModel(item);

if (ModelState.IsValid)

{

using (RationingDbContext db = new RationingDbContext())

{

db.Organizations.Add(item);

db.SaveChanges();

}

}

}

protected void CreateOrganizationForm\_ItemInserted(object sender, FormViewInsertedEventArgs e)

{

if (ModelState.IsValid)

Response.Redirect("~/Tables/Organizations");

}

protected void Unnamed\_Click(object sender, EventArgs e)

{

Response.Redirect("~/Tables/Organizations");

}

}

}

**Products.aspx:**

<%@ Page Title="Виды продукции" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Products.aspx.cs" Inherits="Lab7.Products" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<section>

<div>

<hgroup>

<h2><%: Page.Title %></h2>

</hgroup>

<asp:HyperLink NavigateUrl="~/Tables/ProductsCreate" Text="Добавить новый вид продукции" runat="server" />

<asp:ValidationSummary ShowModelStateErrors="true" runat="server" />

<asp:GridView ID="productView" runat="server" AutoGenerateColumns="False" ShowFooter="True" GridLines="Vertical" CellPadding="4"

ItemType="Lab7.Models.Product" SelectMethod="GetProducts" UpdateMethod="ProductView\_UpdateItem" DeleteMethod="ProductView\_DeleteItem" AutoGenerateEditButton="true" AutoGenerateDeleteButton="true"

CssClass="table table-striped table-bordered" AllowPaging="True">

<Columns>

<asp:BoundField DataField="Id" HeaderText="Id"/>

<asp:BoundField DataField="Name" HeaderText="Название"/>

<asp:BoundField DataField="MeasureUnit" HeaderText="Единица измерения"/>

</Columns>

<PagerSettings Position="TopAndBottom" />

</asp:GridView>

</div>

</section>

</asp:Content>

**Products.aspx.cs:**

using Lab7.Models;

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Web.UI;

namespace Lab7

{

public partial class Products : Page

{

private readonly RationingDbContext \_db = new RationingDbContext();

protected void Page\_Load(object sender, EventArgs e)

{

}

public IOrderedQueryable<Product> GetProducts()

{

IOrderedQueryable<Product> query = \_db.Products.OrderBy(item => item.Id);

return query;

}

public void ProductView\_UpdateItem(int id)

{

Product item = \_db.Products.Find(id);

if (item == null)

{

ModelState.AddModelError("", string.Format("Элемент с идентификатором {0} не был найден", id));

return;

}

TryUpdateModel(item);

if (ModelState.IsValid)

{

\_db.SaveChanges();

}

}

public void ProductView\_DeleteItem(int id)

{

var item = new Product { Id = id };

\_db.Entry(item).State = EntityState.Deleted;

try

{

\_db.SaveChanges();

}

catch (DbUpdateConcurrencyException)

{

ModelState.AddModelError("", string.Format("Item with id {0} no longer exists in the database.", id));

}

}

}

}

**ProductsCreate.aspx.cs:**

<%@ Page Title="Добавление вида продукции" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="ProductsCreate.aspx.cs" Inherits="Lab7.Tables.ProductsCreate" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<h4>Добавление</h4>

<asp:ValidationSummary runat="server" ShowModelStateErrors="true" CssClass="error" />

<asp:FormView runat="server" ID="createProductForm"

ItemType="Lab7.Models.Product"

InsertMethod="CreateProductForm\_InsertItem" DefaultMode="Insert"

RenderOuterTable="false" OnItemInserted="CreateProductForm\_ItemInserted">

<InsertItemTemplate>

Название:

<asp:TextBox ID="NameTextBox" runat="server" Text='<%# Bind("Name") %>' />

<br />

Единица измерения:

<asp:TextBox ID="MeasureUnitTextBox" runat="server" Text='<%# Bind("MeasureUnit") %>' />

<br />

<asp:LinkButton runat="server" CausesValidation="True" CommandName="Insert" Text="Добавить" />

<asp:LinkButton runat="server" CausesValidation="False" CommandName="Cancel" Text="Отмена" OnClick="Unnamed\_Click" />

</InsertItemTemplate>

</asp:FormView>

</asp:Content>

**ProductsCreate.aspx.cs:**

using Lab7.Models;

using System;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Lab7.Tables

{

public partial class ProductsCreate : Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

public void CreateProductForm\_InsertItem()

{

var item = new Product();

TryUpdateModel(item);

if (ModelState.IsValid)

{

using (RationingDbContext db = new RationingDbContext())

{

db.Products.Add(item);

db.SaveChanges();

}

}

}

protected void CreateProductForm\_ItemInserted(object sender, FormViewInsertedEventArgs e)

{

if (ModelState.IsValid)

Response.Redirect("~/Tables/Products");

}

protected void Unnamed\_Click(object sender, EventArgs e)

{

Response.Redirect("~/Tables/Products");

}

}

}

**Ouputs.aspx:**

<%@ Page Title="Выпуск продукции" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Outputs.aspx.cs" Inherits="Lab7.Outputs" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<section>

<div>

<hgroup>

<h2><%: Page.Title %></h2>

</hgroup>

<asp:HyperLink NavigateUrl="~/Tables/OutputsCreate" Text="Добавить новый выпуск продукции" runat="server" />

<asp:ValidationSummary ShowModelStateErrors="true" runat="server" />

<asp:GridView ID="outputView" runat="server" AutoGenerateColumns="False" ShowFooter="True" GridLines="Vertical" CellPadding="4"

ItemType="Lab7.Models.Output" SelectMethod="GetOutputs" UpdateMethod="OutputView\_UpdateItem" DeleteMethod="OutputView\_DeleteItem" AutoGenerateEditButton="true" AutoGenerateDeleteButton="true"

CssClass="table table-striped table-bordered" AllowPaging="True">

<Columns>

<asp:BoundField DataField="Id" HeaderText="Id"/>

<asp:BoundField DataField="OutputPlan" HeaderText="Планируемый выпуск продукции"/>

<asp:BoundField DataField="OutputFact" HeaderText="Фактический выпуск продукции"/>

<asp:BoundField DataField="Quarter" HeaderText="Квартал" />

<asp:BoundField DataField="Year" HeaderText="Год"/>

<asp:TemplateField HeaderText="Организация">

<EditItemTemplate>

<asp:DropDownList ID="OrganizationIdDropDownList" runat="server" SelectMethod="GetOrganizations" DataTextField="Name" DataValueField="Id" SelectedValue='<%# Bind("OrganizationId") %>'>

</asp:DropDownList>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="OrganizationId" runat="server" Text='<%# Eval("Organization.Name") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Вид продукции">

<EditItemTemplate>

<asp:DropDownList ID="ProductionIdDropDownList" runat="server" SelectMethod="GetProducts" DataTextField="Name" DataValueField="Id" SelectedValue='<%# Bind("ProductId") %>'>

</asp:DropDownList>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="ProductId" runat="server" Text='<%# Eval("Product.Name") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

</Columns>

<PagerSettings Position="TopAndBottom" />

</asp:GridView>

</div>

</section>

</asp:Content>

**Outputs.aspx.cs:**

using Lab7.Models;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Web.UI;

namespace Lab7

{

public partial class Outputs : Page

{

private readonly RationingDbContext \_db = new RationingDbContext();

protected void Page\_Load(object sender, EventArgs e)

{

}

public IOrderedQueryable<Output> GetOutputs()

{

IOrderedQueryable<Output> query = \_db.Outputs.OrderBy(item => item.Id);

return query;

}

public void OutputView\_UpdateItem(int id)

{

Output item = \_db.Outputs.Find(id);

if (item == null)

{

ModelState.AddModelError("", string.Format("Элемент с идентификатором {0} не был найден", id));

return;

}

TryUpdateModel(item);

if (ModelState.IsValid)

{

\_db.SaveChanges();

}

}

public void OutputView\_DeleteItem(int id)

{

var item = new Output { Id = id };

\_db.Entry(item).State = EntityState.Deleted;

try

{

\_db.SaveChanges();

}

catch (DbUpdateConcurrencyException)

{

ModelState.AddModelError("", string.Format("Item with id {0} no longer exists in the database.", id));

}

}

public IOrderedQueryable<Organization> GetOrganizations()

{

IOrderedQueryable<Organization> query = \_db.Organizations.OrderBy(item => item.Id);

return query;

}

public IOrderedQueryable<Product> GetProducts()

{

IOrderedQueryable<Product> query = \_db.Products.OrderBy(item => item.Id);

return query;

}

}

}

**OutputsCreate.aspx:**

<%@ Page Title="Добавление выпуска продукции" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="OutputsCreate.aspx.cs" Inherits="Lab7.Tables.OutputsCreate" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<h4>Добавление</h4>

<asp:ValidationSummary runat="server" ShowModelStateErrors="true" CssClass="error" />

<asp:FormView runat="server" ID="createOutputForm"

ItemType="Lab7.Models.Output"

InsertMethod="OutputForm\_InsertItem" DefaultMode="Insert"

RenderOuterTable="false" OnItemInserted="CreateOutputForm\_ItemInserted">

<InsertItemTemplate>

Планируемый выпуск продукции:

<asp:TextBox ID="OutputPlanTextBox" runat="server" Text='<%# Bind("OutputPlan") %>' />

<br />

Фактический выпуск продукции:

<asp:TextBox ID="OutputFactTextBox" runat="server" Text='<%# Bind("OutputFact") %>' />

<br />

Квартал:

<asp:TextBox ID="QuarterTextBox" runat="server" Text='<%# Bind("Quarter") %>' />

<br />

Год:

<asp:TextBox ID="YearTextBox" runat="server" Text='<%# Bind("Year") %>' />

<br />

Организация:

<asp:DropDownList ID="OrganizationIdDropDownList" SelectMethod="GetOrganizations" runat="server" DataTextField="Name" DataValueField="Id" SelectedValue='<%# Bind("OrganizationId") %>'>

</asp:DropDownList>

<br />

Вид продукции:

<asp:DropDownList ID="ProductionIdDropDownList" runat="server" SelectMethod="GetProducts" DataTextField="Name" DataValueField="Id" SelectedValue='<%# Bind("ProductId") %>'>

</asp:DropDownList>

<br />

<asp:LinkButton runat="server" CausesValidation="True" CommandName="Insert" Text="Добавить" />

<asp:LinkButton runat="server" CausesValidation="False" CommandName="Cancel" Text="Отмена" OnClick="Unnamed\_Click" />

</InsertItemTemplate>

</asp:FormView>

</asp:Content>

**OutputsCreate.aspx.cs:**

using Lab7.Models;

using System;

using System.Linq;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Lab7.Tables

{

public partial class OutputsCreate : Page

{

private readonly RationingDbContext \_db = new RationingDbContext();

protected void Page\_Load(object sender, EventArgs e)

{

}

public void OutputForm\_InsertItem()

{

var item = new Output();

TryUpdateModel(item);

if (ModelState.IsValid)

{

\_db.Outputs.Add(item);

\_db.SaveChanges();

}

}

protected void CreateOutputForm\_ItemInserted(object sender, FormViewInsertedEventArgs e)

{

if (ModelState.IsValid)

Response.Redirect("~/Tables/Outputs");

}

protected void Unnamed\_Click(object sender, EventArgs e)

{

Response.Redirect("~/Tables/Outputs");

}

public IOrderedQueryable<Organization> GetOrganizations()

{

IOrderedQueryable<Organization> query = \_db.Organizations.OrderBy(item => item.Id);

return query;

}

public IOrderedQueryable<Product> GetProducts()

{

IOrderedQueryable<Product> query = \_db.Products.OrderBy(item => item.Id);

return query;

}

}

}

**Realisations.aspx:**

<%@ Page Title="Реализация продукции" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Realisations.aspx.cs" Inherits="Lab7.Realisations" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<section>

<div>

<hgroup>

<h2><%: Page.Title %></h2>

</hgroup>

<asp:HyperLink NavigateUrl="~/Tables/RealisationsCreate" Text="Добавить новую реализацию продукции" runat="server" />

<asp:ValidationSummary ShowModelStateErrors="true" runat="server" />

<asp:GridView ID="realisationRateView" runat="server" AutoGenerateColumns="False" ShowFooter="True" GridLines="Vertical" CellPadding="4"

ItemType="Lab7.Models.Realisation" SelectMethod="GetRealisations" UpdateMethod="RealisationView\_UpdateItem" DeleteMethod="RealisationView\_DeleteItem" AutoGenerateEditButton="true" AutoGenerateDeleteButton="true"

CssClass="table table-striped table-bordered" AllowPaging="True">

<Columns>

<asp:BoundField DataField="Id" HeaderText="Id"/>

<asp:BoundField DataField="RealisationPlan" HeaderText="Планируемая реализация продукции"/>

<asp:BoundField DataField="RealisationFact" HeaderText="Фактическая реализация продукции"/>

<asp:BoundField DataField="Quarter" HeaderText="Квартал" />

<asp:BoundField DataField="Year" HeaderText="Год"/>

<asp:TemplateField HeaderText="Организация">

<EditItemTemplate>

<asp:DropDownList ID="OrganizationIdDropDownList" runat="server" SelectMethod="GetOrganizations" DataTextField="Name" DataValueField="Id" SelectedValue='<%# Bind("OrganizationId") %>'>

</asp:DropDownList>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="OrganizationId" runat="server" Text='<%# Eval("Organization.Name") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Вид продукции">

<EditItemTemplate>

<asp:DropDownList ID="ProductionIdDropDownList" runat="server" SelectMethod="GetProducts" DataTextField="Name" DataValueField="Id" SelectedValue='<%# Bind("ProductId") %>'>

</asp:DropDownList>

</EditItemTemplate>

<ItemTemplate>

<asp:Label ID="ProductId" runat="server" Text='<%# Eval("Product.Name") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

</Columns>

<PagerSettings Position="TopAndBottom" />

</asp:GridView>

</div>

</section>

</asp:Content>

**Realisations.aspx.cs:**

using Lab7.Models;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Web.UI;

namespace Lab7

{

public partial class Realisations : Page

{

private readonly RationingDbContext \_db = new RationingDbContext();

protected void Page\_Load(object sender, EventArgs e)

{

}

public IOrderedQueryable<ou> GetRealisations()

{

IOrderedQueryable<Realisation> query = \_db.Realisations.OrderBy(item => item.Id);

return query;

}

public void RealsiationView\_UpdateItem(int id)

{

Realisation item = \_db.Realisations.Find(id);

if (item == null)

{

ModelState.AddModelError("", string.Format("Элемент с идентификатором {0} не был найден", id));

return;

}

TryUpdateModel(item);

if (ModelState.IsValid)

{

\_db.SaveChanges();

}

}

public void RealisationView\_DeleteItem(int id)

{

var item = new Realisation { Id = id };

\_db.Entry(item).State = EntityState.Deleted;

try

{

\_db.SaveChanges();

}

catch (DbUpdateConcurrencyException)

{

ModelState.AddModelError("", string.Format("Item with id {0} no longer exists in the database.", id));

}

}

public IOrderedQueryable<Organization> GetOrganizations()

{

IOrderedQueryable<Organization> query = \_db.Organizations.OrderBy(item => item.Id);

return query;

}

public IOrderedQueryable<Product> GetProducts()

{

IOrderedQueryable<Product> query = \_db.Products.OrderBy(item => item.Id);

return query;

}

}

}

**RealisationsCreate.aspx:**

<%@ Page Title="Добавление реализации продукции" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="RealisationsCreate.aspx.cs" Inherits="Lab7.Tables.RealisationsCreate" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

<h4>Добавление</h4>

<asp:ValidationSummary runat="server" ShowModelStateErrors="true" CssClass="error" />

<asp:FormView runat="server" ID="createRealisationForm"

ItemType="Lab7.Models.Realisation"

InsertMethod="CreateRealisationForm\_InsertItem" DefaultMode="Insert"

RenderOuterTable="false" OnItemInserted="CreateRealisationForm\_ItemInserted">

<InsertItemTemplate>

Планируемая реализация продукции:

<asp:TextBox ID="RealisationPlanTextBox" runat="server" Text='<%# Bind("RealisationPlan") %>' />

<br />

Фактическая реализация продукции:

<asp:TextBox ID="RealisationFactTextBox" runat="server" Text='<%# Bind("RealisationFact") %>' />

<br />

Квартал:

<asp:TextBox ID="QuarterTextBox" runat="server" Text='<%# Bind("Quarter") %>' />

<br />

Год:

<asp:TextBox ID="YearTextBox" runat="server" Text='<%# Bind("Year") %>' />

<br />

Организация:

<asp:DropDownList ID="OrganizationIdDropDownList" SelectMethod="GetOrganizations" runat="server" DataTextField="Name" DataValueField="Id" SelectedValue='<%# Bind("OrganizationId") %>'>

</asp:DropDownList>

<br />

Вид продукции:

<asp:DropDownList ID="ProductionIdDropDownList" runat="server" SelectMethod="GetProducts" DataTextField="Name" DataValueField="Id" SelectedValue='<%# Bind("ProductId") %>'>

</asp:DropDownList>

<br />

<asp:LinkButton runat="server" CausesValidation="True" CommandName="Insert" Text="Добавить" />

<asp:LinkButton runat="server" CausesValidation="False" CommandName="Cancel" Text="Отмена" OnClick="Unnamed\_Click" />

</InsertItemTemplate>

</asp:FormView>

</asp:Content>

**RealisationsCreate.aspx.cs:**

using Lab7.Models;

using System;

using System.Linq;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Lab7.Tables

{

public partial class RealisationsCreate : Page

{

private readonly RationingDbContext \_db = new RationingDbContext();

protected void Page\_Load(object sender, EventArgs e)

{

}

public void CreateRealisationForm\_InsertItem()

{

var item = new Realisation();

TryUpdateModel(item);

if (ModelState.IsValid)

{

\_db.Realisations.Add(item);

\_db.SaveChanges();

}

}

protected void CreateRealisationsForm\_ItemInserted(object sender, FormViewInsertedEventArgs e)

{

if (ModelState.IsValid)

Response.Redirect("~/Tables/Realisations");

}

protected void Unnamed\_Click(object sender, EventArgs e)

{

Response.Redirect("~/Tables/Realisations");

}

public IOrderedQueryable<Organization> GetOrganizations()

{

IOrderedQueryable<Organization> query = \_db.Organizations.OrderBy(item => item.Id);

return query;

}

public IOrderedQueryable<Product> GetProducts()

{

IOrderedQueryable<Product> query = \_db.Products.OrderBy(item => item.Id);

return query;

}

}

}