Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа №5**

**по дисциплине**

**«Технологии программирования»**

**ШАБЛОН ПРОЕКТИРОВАНИЯ MVC В ASP.NET**

**Выполнил**:

ст. гр. ПРИ-120

Д. А. Грачев

**Принял**:

Данилов В. В.

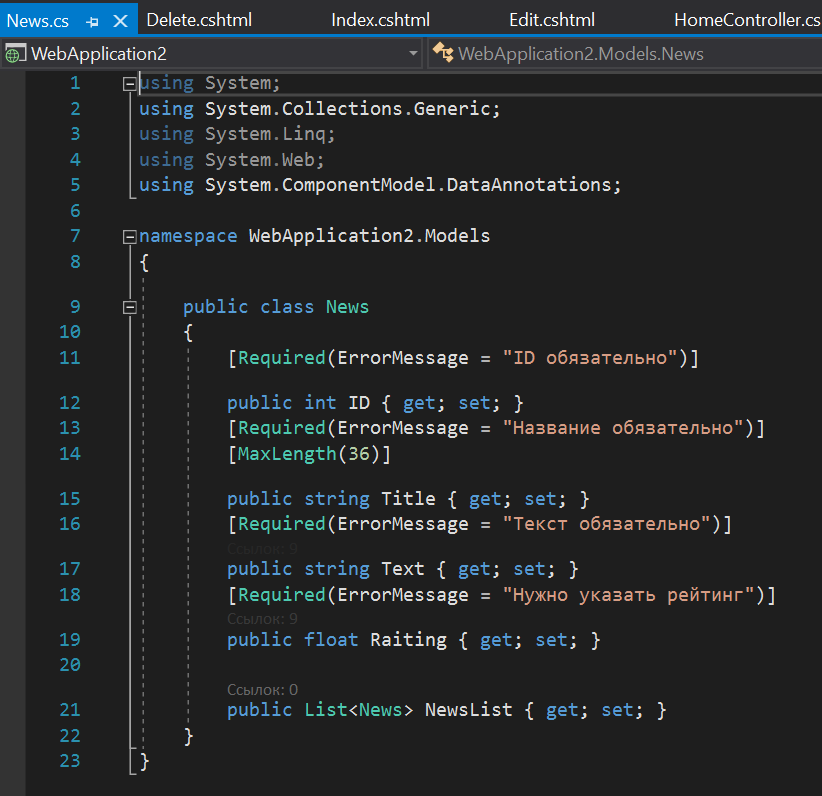
Владимир, 2021

ЦЕЛЬ РАБОТЫ

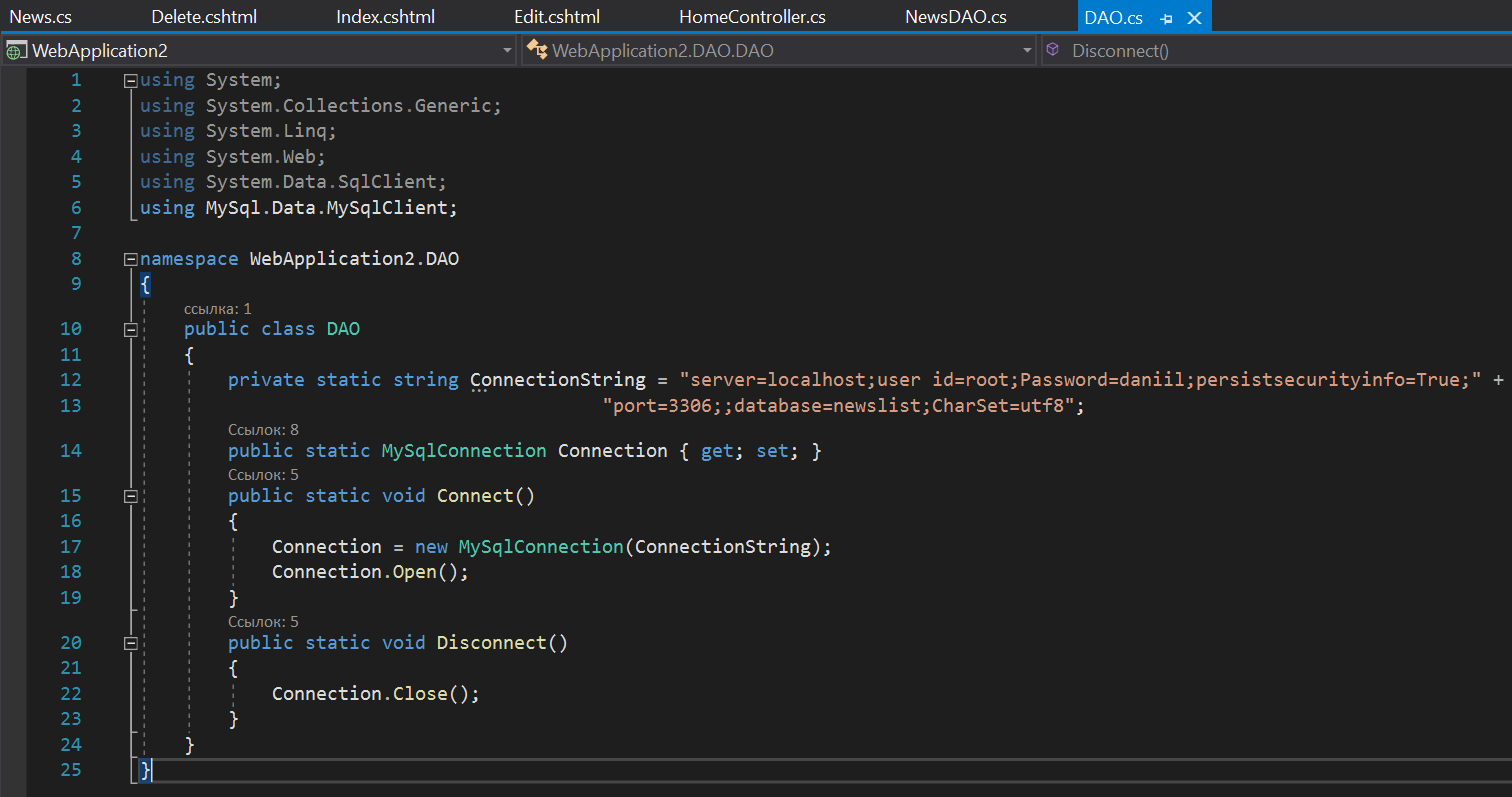
Познакомиться с технологией создания и использования шаблона проектирования MVC на платформе ASP.NET.

ВЫПОЛНЕНИЕ РАБОТЫ

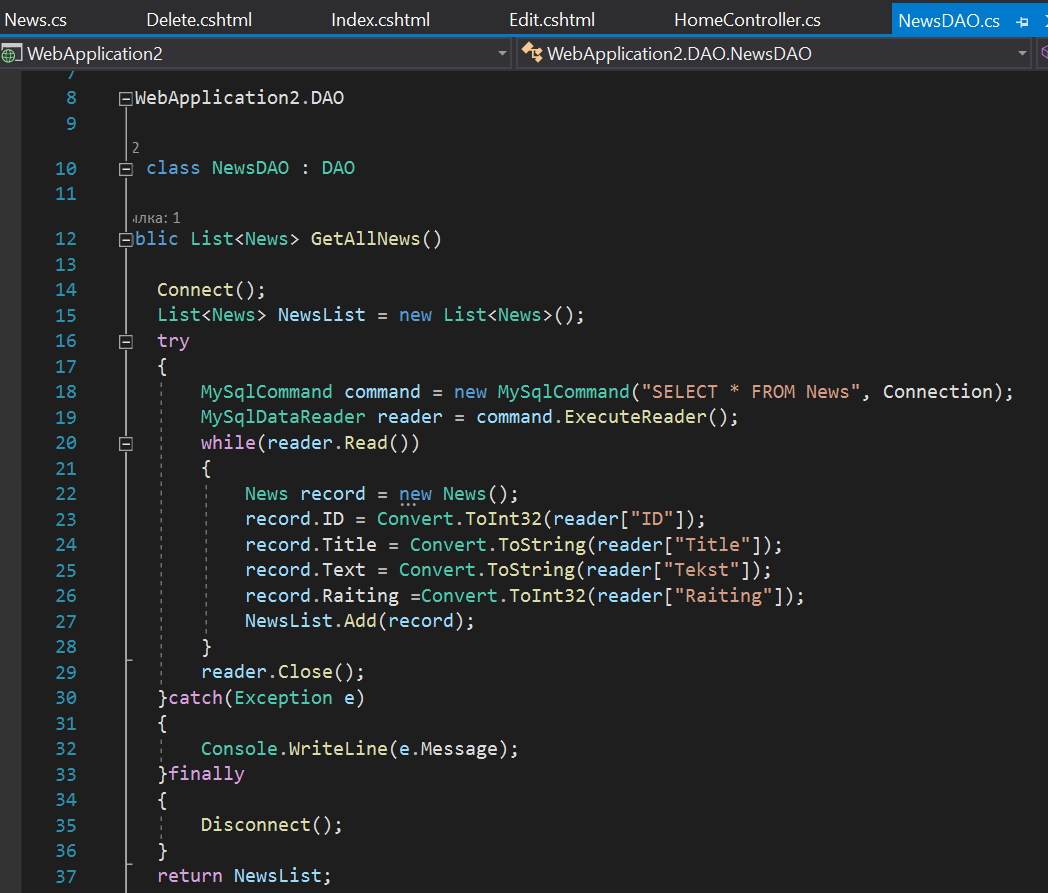
1. Создадим модель News.cs



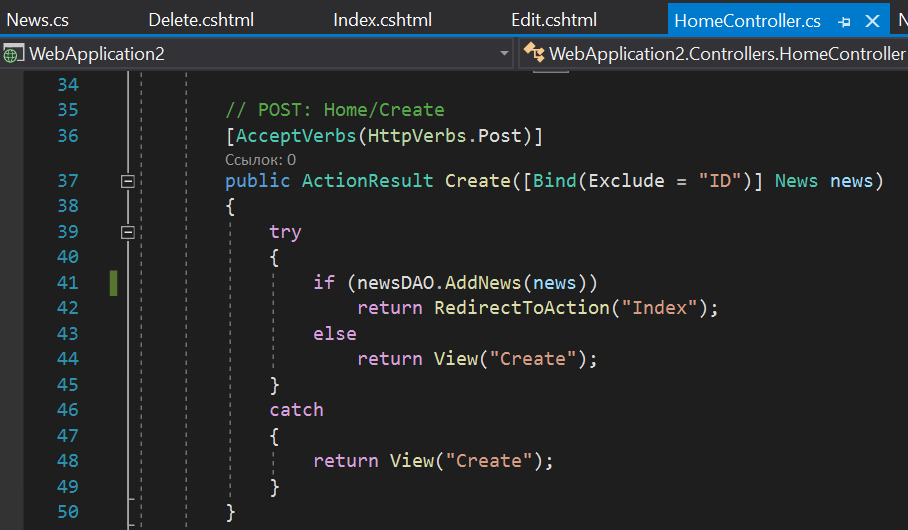
1. Создадим класс DAO.cs



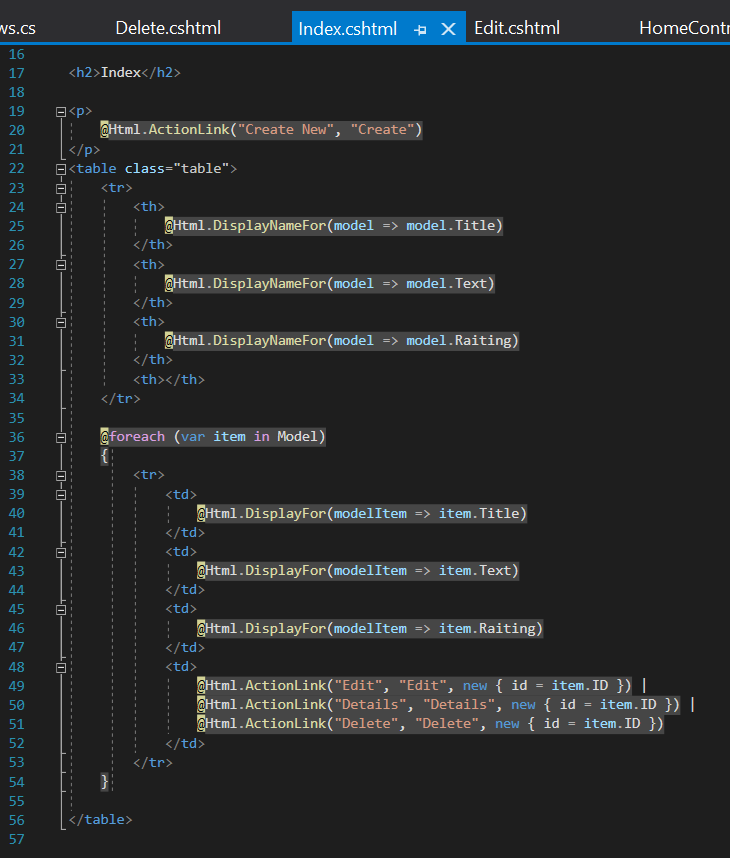
1. Создадим класс NewsDAO и пропишем сразу метод получения всех новостей из бд



1. Создадим HomeController и пропишем обработчик создания новости



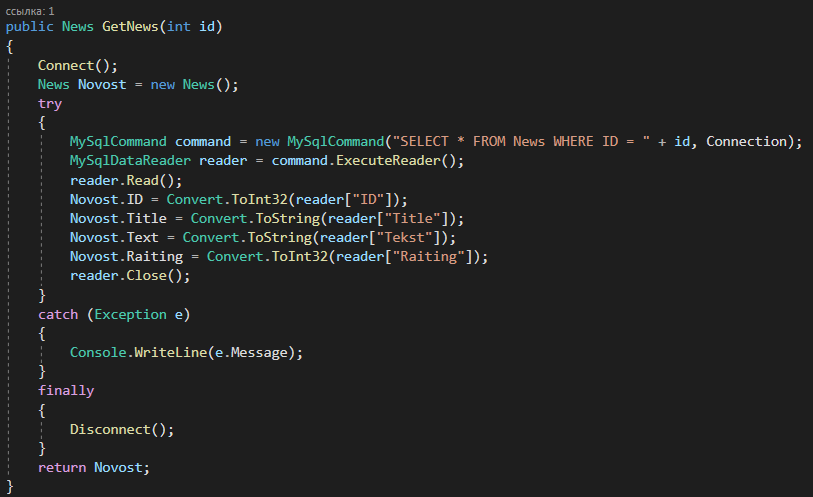
1. Создадим представление Index



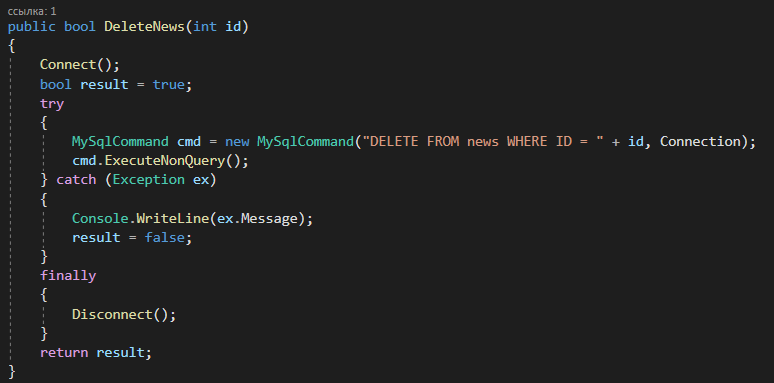
1. Функционал добавления записи



1. Функционал полного описания



1. Функционал удаления



1. Функционал изменения



Листинг программы:

DAO.cs

namespace WebApplication2.DAO

{

    public class DAO

    {

        private static string ConnectionString = "server=localhost;user id=root;Password=daniil;persistsecurityinfo=True;" +

                                          "port=3306;;database=newslist;CharSet=utf8";

        public static MySqlConnection Connection { get; set; }

        public static void Connect()

        {

            Connection = new MySqlConnection(ConnectionString);

            Connection.Open();

        }

        public static void Disconnect()

        {

            Connection.Close();

        }

    }

}

News.cs

namespace WebApplication2.DAO

{

    public class NewsDAO : DAO

    {

        public List<News> GetAllNews()

        {

            Connect();

            List<News> NewsList = new List<News>();

            try

            {

                MySqlCommand command = new MySqlCommand("SELECT \* FROM News", Connection);

                MySqlDataReader reader = command.ExecuteReader();

                while(reader.Read())

                {

                    News record = new News();

                    record.ID = Convert.ToInt32(reader["ID"]);

                    record.Title = Convert.ToString(reader["Title"]);

                    record.Text = Convert.ToString(reader["Tekst"]);

                    record.Raiting =Convert.ToInt32(reader["Raiting"]);

                    NewsList.Add(record);

                }

                reader.Close();

            }catch(Exception e)

            {

                Console.WriteLine(e.Message);

            }finally

            {

                Disconnect();

            }

            return NewsList;

        }

        public News GetNews(int id)

        {

            Connect();

            News Novost = new News();

            try

            {

                MySqlCommand command = new MySqlCommand("SELECT \* FROM News WHERE ID = " + id, Connection);

                MySqlDataReader reader = command.ExecuteReader();

                reader.Read();

                Novost.ID = Convert.ToInt32(reader["ID"]);

                Novost.Title = Convert.ToString(reader["Title"]);

                Novost.Text = Convert.ToString(reader["Tekst"]);

                Novost.Raiting = Convert.ToInt32(reader["Raiting"]);

                reader.Close();

            }

            catch (Exception e)

            {

                Console.WriteLine(e.Message);

            }

            finally

            {

                Disconnect();

            }

            return Novost;

        }

        public bool AddNews(News news)

        {

            bool result = true;

            Connect();

            try

            {

                MySqlCommand cmd = new MySqlCommand("INSERT News(title, tekst, raiting) VALUES (@title, @tekst, @raiting)",Connection);

                cmd.Parameters.Add(new MySqlParameter("@title",news.Title));

                cmd.Parameters.Add(new MySqlParameter("@tekst",news.Text));

                cmd.Parameters.Add(new MySqlParameter("@raiting",news.Raiting));

                cmd.ExecuteNonQuery();

            }

            catch (Exception)

            {

                result = false;

            }

            finally

            {

                Disconnect();

            }

            return result;

        }

        public bool EditNews(News news)

        {

            bool result = true;

            Connect();

            try

            {

                MySqlCommand cmd = new MySqlCommand("UPDATE News SET " +

                    "title = @title, tekst = @text, raiting = @raiting " +

                    "WHERE ID = 4", Connection);

                cmd.Parameters.Add(new MySqlParameter("@title", news.Title));

                cmd.Parameters.Add(new MySqlParameter("@text", news.Text));

                cmd.Parameters.Add(new MySqlParameter("@raiting", news.Raiting));

                cmd.ExecuteNonQuery();

            }

            catch (Exception ex)

            {

                result = false;

                Console.WriteLine(ex.Message);

            }

            finally

            {

                Disconnect();

            }

            return result;

        }

        public bool DeleteNews(int id)

        {

            Connect();

            bool result = true;

            try

            {

                MySqlCommand cmd = new MySqlCommand("DELETE FROM news WHERE ID = " + id, Connection);

                cmd.ExecuteNonQuery();

            } catch (Exception ex)

            {

                Console.WriteLine(ex.Message);

                result = false;

            }

            finally

            {

                Disconnect();

            }

            return result;

        }

    }

}

HomeController.cs

namespace WebApplication2.Controllers

{

    public class HomeController : Controller

    {

        NewsDAO newsDAO = new NewsDAO();

*// GET: Home*

        public ActionResult Index()

        {

            return View(newsDAO.GetAllNews());

        }

*// GET: Home/Details/5*

        public ActionResult Details(int id)

        {

            return View(newsDAO.GetNews(id));

        }

*// GET: Home/Create*

        public ActionResult Create()

        {

            return View();

        }

*// POST: Home/Create*

        [AcceptVerbs(HttpVerbs.Post)]

        public ActionResult Create([Bind(Exclude = "ID")] News news)

        {

            try

            {

                if (newsDAO.AddNews(news))

                    return RedirectToAction("Index");

                else

                    return View("Create");

            }

            catch

            {

                return View("Create");

            }

        }

*// GET: Home/Edit/5*

        public ActionResult Edit(int id)

        {

            return View();

        }

*// POST: Home/Edit/5*

        [HttpPost]

        public ActionResult Edit(News news)

        {

            try

            {

*// TODO: Add update logic here*

*//View(newsDAO.GetNews(news.ID));*

                if (newsDAO.EditNews(news))

                {

                    return RedirectToAction("Index");

                } else

                {

                    return View("Edit");

                }

            }

            catch

            {

                return View("Edit");

            }

        }

*// GET: Home/Delete/5*

        public ActionResult Delete(int id)

        {

            return View();

        }

*// POST: Home/Delete/5*

        [HttpPost]

        public ActionResult Delete(int id, FormCollection collection)

        {

            try

            {

                if (newsDAO.DeleteNews(id))

                {

                    return RedirectToAction("Index");

                } else

                {

                    return View("Delete");

                }

            }

            catch

            {

                return View();

            }

        }

    }

}

Index.cshtml

@{

    ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

    @Html.ActionLink("Create New", "Create")

</p>

<table class="table">

    <tr>

        <th>

            @Html.DisplayNameFor(model => model.Title)

        </th>

        <th>

            @Html.DisplayNameFor(model => model.Text)

        </th>

        <th>

            @Html.DisplayNameFor(model => model.Raiting)

        </th>

        <th></th>

    </tr>

    @foreach (var item in Model)

    {

        <tr>

            <td>

                @Html.DisplayFor(modelItem => item.Title)

            </td>

            <td>

                @Html.DisplayFor(modelItem => item.Text)

            </td>

            <td>

                @Html.DisplayFor(modelItem => item.Raiting)

            </td>

            <td>

                @Html.ActionLink("Edit", "Edit", new { id = item.ID }) |

                @Html.ActionLink("Details", "Details", new { id = item.ID }) |

                @Html.ActionLink("Delete", "Delete", new { id = item.ID })

            </td>

        </tr>

    }

</table>

Create.cshtml

@model WebApplication2.Models.News

@{

    ViewBag.Title = "Create";

}

<h2>Create</h2>

@using (Html.BeginForm())

{

    @Html.AntiForgeryToken()

    <div class="form-horizontal">

        <h4>News</h4>

        <hr />

        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

        <div class="form-group">

            @Html.LabelFor(model => model.Title, htmlAttributes: new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Title, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Title, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.Text, htmlAttributes: new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Text, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Text, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.Raiting, htmlAttributes: new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Raiting, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Raiting, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

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

                <input type="submit" value="Create" class="btn btn-default" />

            </div>

        </div>

    </div>

}

<div>

    @Html.ActionLink("Back to List", "Index")

</div>

Delete.cshtml

@model WebApplication2.Models.News

@{

    ViewBag.Title = "Delete";

}

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>

<div>

    @using (Html.BeginForm()) {

        @Html.AntiForgeryToken()

        <div class="form-actions no-color">

            <input type="submit" value="Delete" class="btn btn-default" /> |

            @Html.ActionLink("Back to List", "Index")

        </div>

    }

</div>

Details.cshtml

@model WebApplication2.Models.News

@{

    ViewBag.Title = "Details";

}

<h2>Details</h2>

<div>

    <h4>News</h4>

    <hr />

    <dl class="dl-horizontal">

        <dt>

            @Html.DisplayNameFor(model => model.Title)

        </dt>

        <dd>

            @Html.DisplayFor(Model => Model.Title)

        </dd>

        <dt>

            @Html.DisplayNameFor(model => model.Text)

        </dt>

        <dd>

            @Html.DisplayFor(Model => Model.Text)

        </dd>

        <dt>

            @Html.DisplayNameFor(model => model.Raiting)

        </dt>

        <dd>

            @Html.DisplayFor(Model => Model.Raiting)

        </dd>

    </dl>

</div>

<p>

    @Html.ActionLink("Back to List", "Index")

</p>

Edit.cshtml

@model WebApplication2.Models.News

@{

    ViewBag.Title = "Edit";

}

<h2>Edit</h2>

@using (Html.BeginForm())

{

    @Html.AntiForgeryToken()

    <div class="form-horizontal">

        <h4>News</h4>

        <hr />

        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

        @Html.HiddenFor(model => model.ID)

        <div class="form-group">

            @Html.LabelFor(model => model.Title, htmlAttributes: new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Title, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Title, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.Text, htmlAttributes: new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Text, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Text, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

            @Html.LabelFor(model => model.Raiting, htmlAttributes: new { @class = "control-label col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.Raiting, new { htmlAttributes = new { @class = "form-control" } })

                @Html.ValidationMessageFor(model => model.Raiting, "", new { @class = "text-danger" })

            </div>

        </div>

        <div class="form-group">

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

                <input type="submit" value="Save" class="btn btn-default" />

            </div>

        </div>

    </div>

}

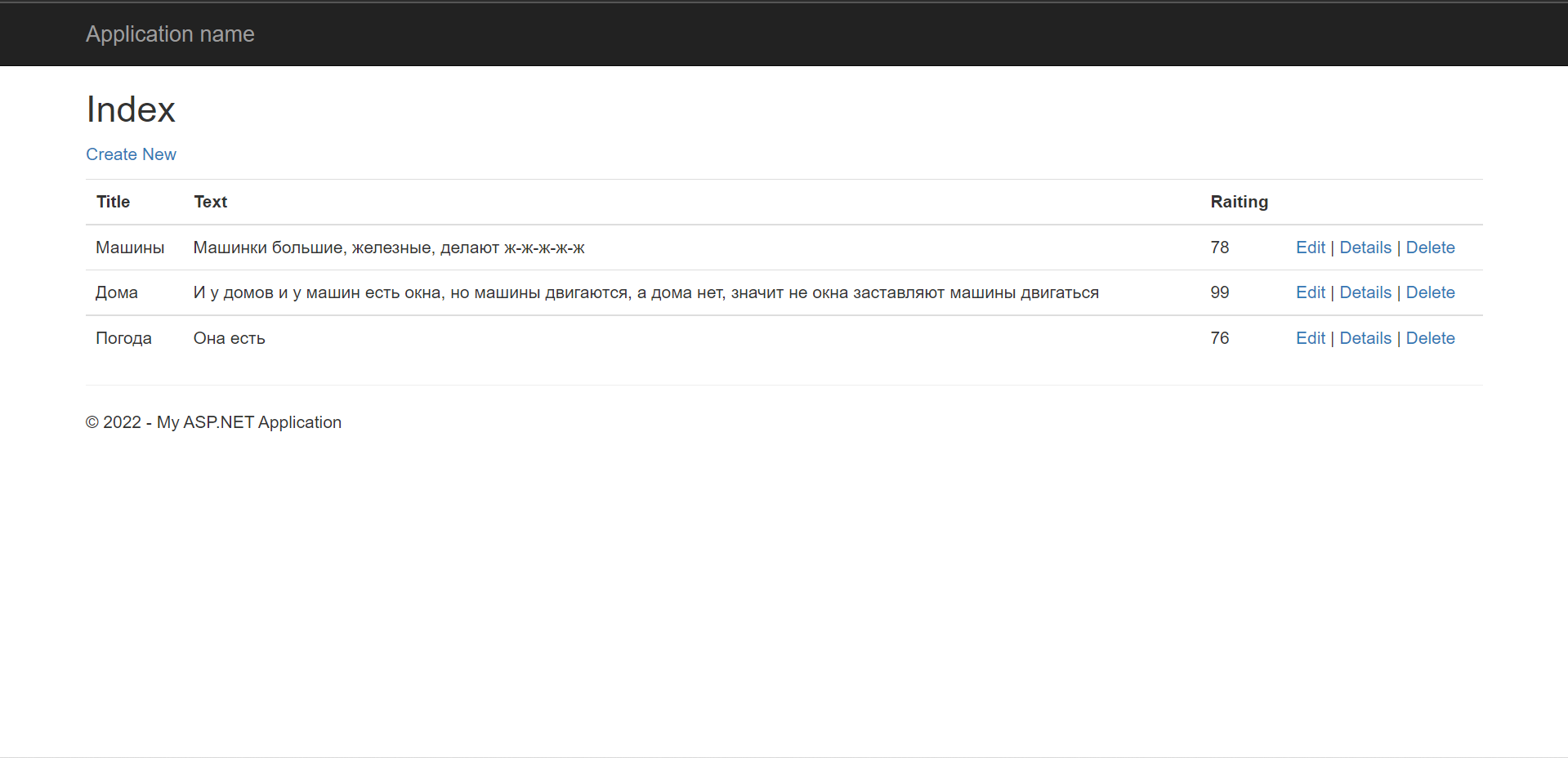
<div>

    @Html.ActionLink("Back to List", "Index")

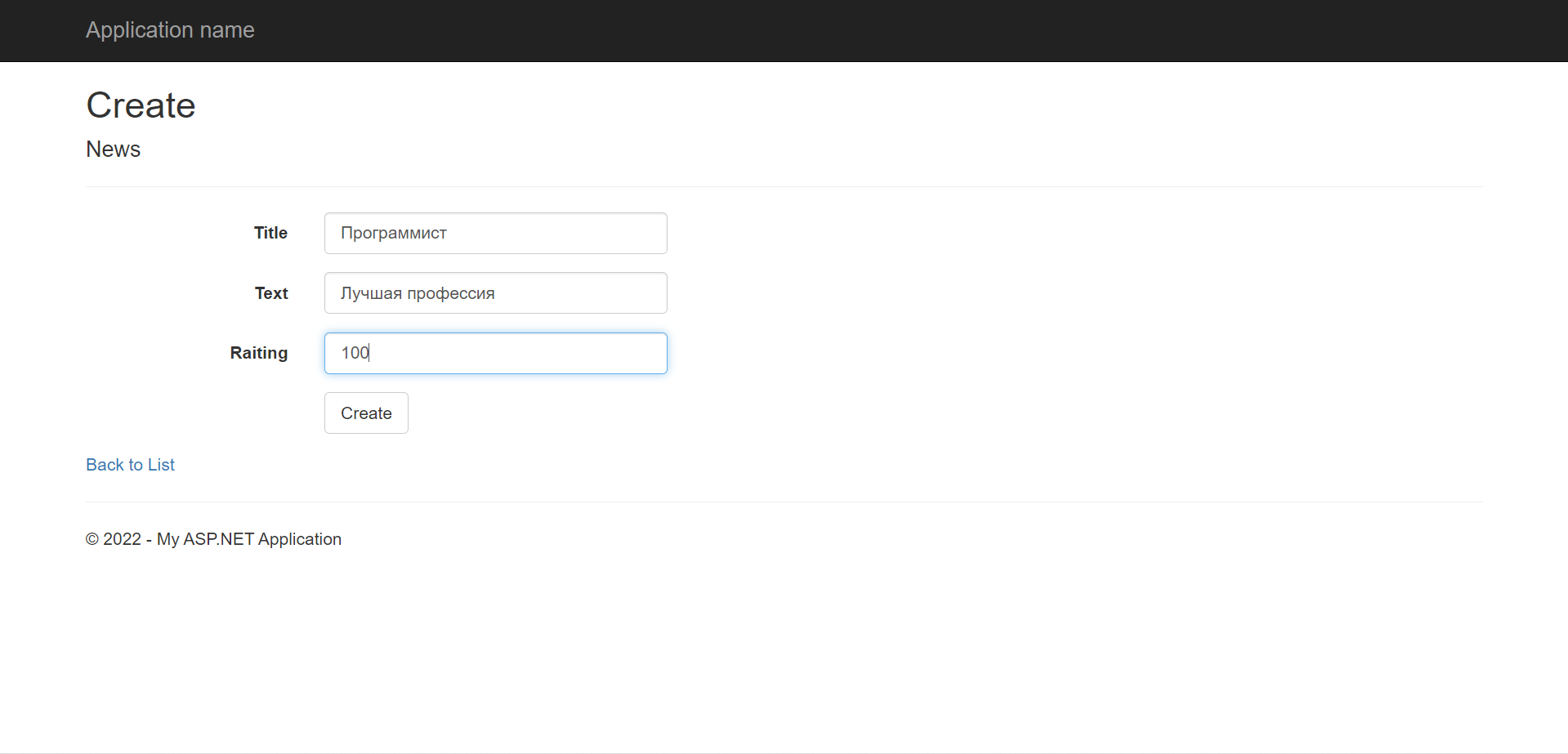
</div>

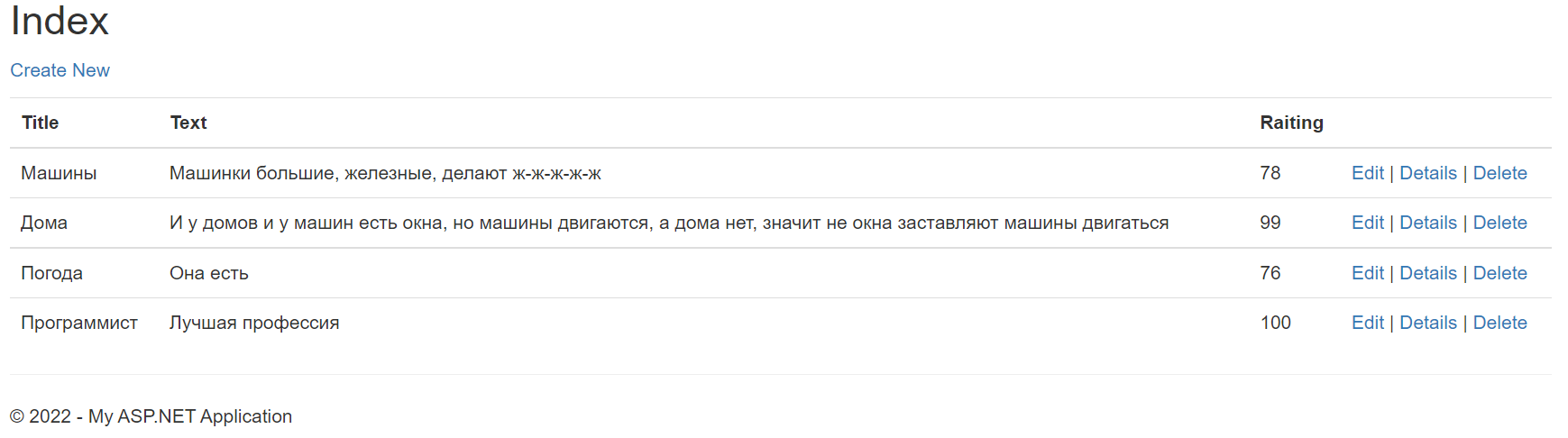
Скриншот работы программы:

Стартовая страница:

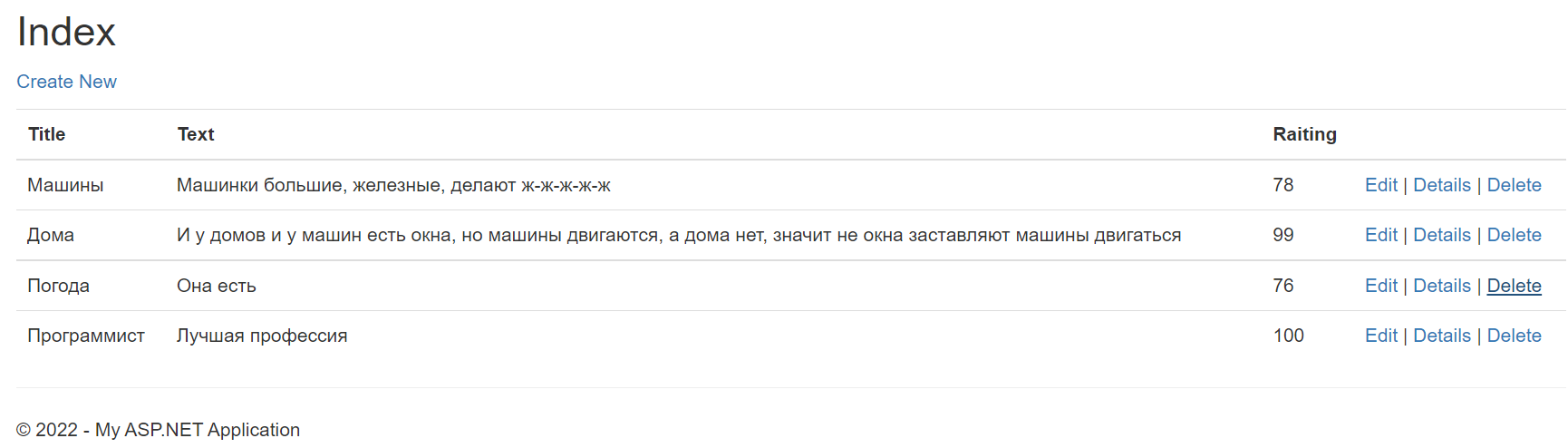


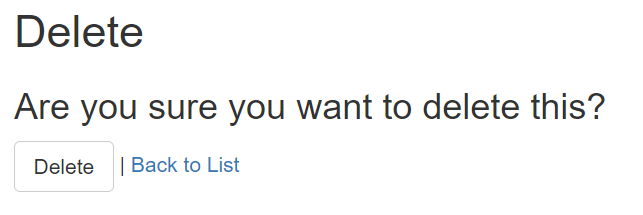
Страница добавления:

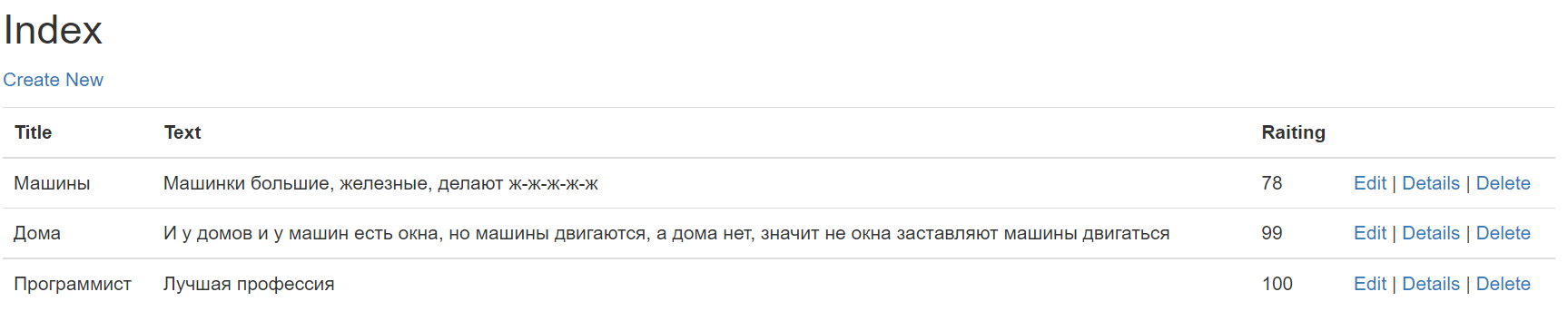




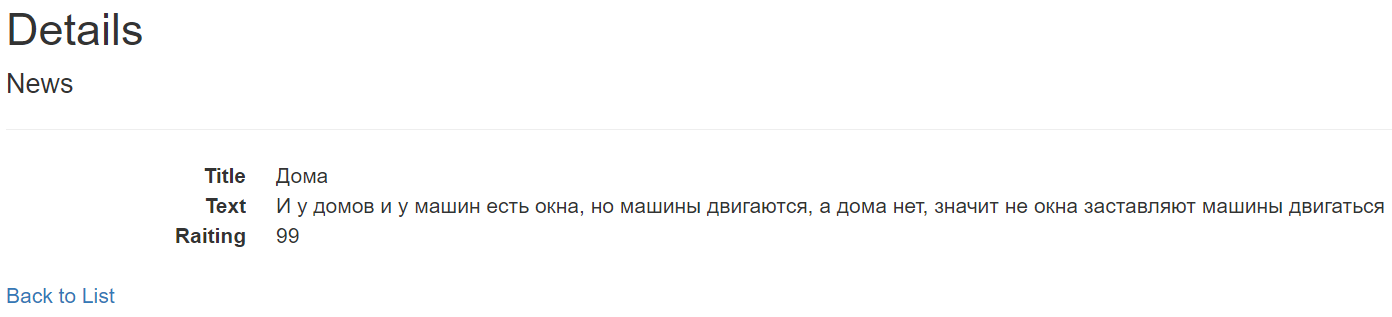
Страница удаления:



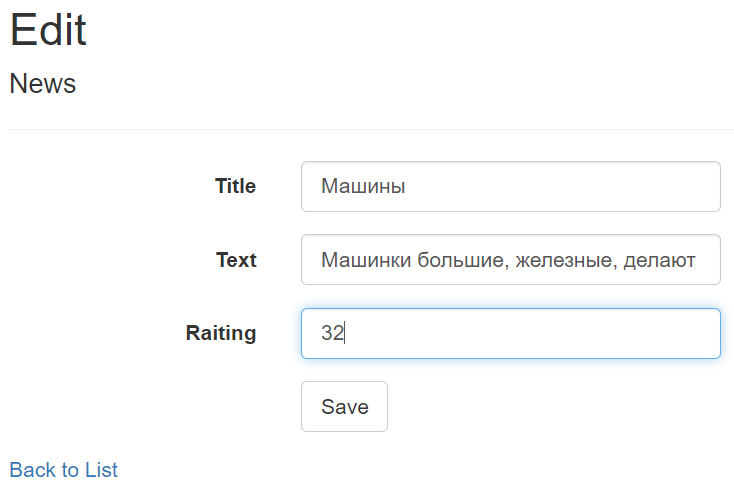




Детализация:



Изменение:



ВЫВОД

В ходе выполнения лабораторной работы познакомился с технологией создания и использования шаблона проектирования MVC на платформе ASP.NET.