**Министерство образования и науки Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ»**

**Факультет «ИКТ»**

**Дисциплина**

«Создание программного обеспечения инфокоммуникационных систем»

**Лабораторная работа 5**

**«**Создание Web-приложения**»**

**Выполнил:**

студент группы K3415

Лобченко Д. Р.

**Проверил**:

Осипов Н. А.

Санкт-Петербург

2021

**Задача, решаемая в лабораторной работе.**

Требуется создать веб-сайт, который дал бы возможность приглашать на семинар, а приглашенным отправлять ответы на приглашение.

Должны быть следующие основные средства:

* страница, которая отображает информацию о семинаре и форму RSVP;
* проверка достоверности для формы RSVP, которая будет отображать

страницу подтверждения;

* страница, на которой выводится список ответов от приглашенных.

**Лабораторная работа 1. Создание основы веб-сайта приглашения на семинар**

**Упражнение 1. Создание основы веб-сайта ASP.NET**

1. Создадим новый проект ASP.NET с именем «RSVP».
2. Создадим модель данных. Для этого добавим в проект новый класс с именем **GuestResponse** и определим автоматически реализуемые свойства в соответствии с заданием, а также два конструктора (базовый и переопределенный, принимающий имя, электронный адрес, телефон и информацию о согласии).
3. Определим временное хранилище данных. Для этого создадим новый класс с именем **ResponseRepository** и отредактируем, добавив логику получения и добавления объектов данных в коллекцию.

**Лабораторная работа 1. Создание основы веб-сайта приглашения на семинар**

**Упражнение 2. Создание форм веб-сайта**

1. Добавим в проект новую форму с именем **Start.aspx** и отредактируем ее содержимое, чтобы выводилось приглашение на семинар.
2. Зададим созданную форму в качестве главной страницы для нашего приложения.
3. Запустим и протестируем приложение.
4. Добавим в проект новую форму с именем **Reg.aspx** и отредактируем ее содержимое, чтобы выводилась форма регистрации участника на семинар.
5. Откроем созданную страницу в браузере, чтобы протестировать корректное отображение всех элементов.
6. Добавим в приложение таблицу стилей **Styles**. И определим стили для элементов пользовательского интерфейса.
7. Добавим ссылку на таблицу стилей на страницах нашего приложения, чтобы применить созданные стили.
8. Проверим влияние стилей на отображение пользовательских элементов в браузере.

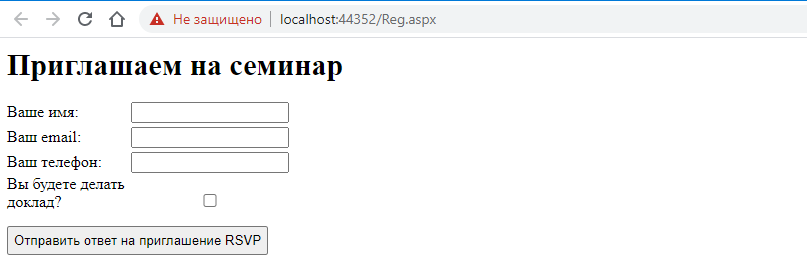


Рисунок 3 - результат применения стилей к странице регистрации участика.

**Лабораторная работа 2. Обработка событий формы**

**Упражнение 1. Обработка события загрузки формы**

1. В обработчике события загрузки формы определим, относится ли запрос, на который производится ответ, к форме, отправленной обратно серверу. Если да, то создадим новый экземпляр объекта GuestResponse модели данных с полученными от элементов формы данными.
2. Добавим в созданный объект информацию из запроса.
3. Для реагирования на данное событие отправки формы перенаправим пользователя на соответствующие вариантам выбора страницы.

**Лабораторная работа 2. Обработка событий формы**

**Упражнение 2. Создание HTML-файлов**

1. Создадим две html страницы, перенаправление на которые было определено в предыдущем упражнении.
2. Отредактируем содержимое данных страниц в соответствии с заданием.
3. Протестируем приложение.

**Лабораторная работа 3. Применение операторов C# в коде страницы**

**Упражнение 1. Создание итогового представления**

1. Добавим в приложение новую форму Summary.aspx для отображения информации о пользователях, выступающих на семинаре.
2. Отредактируем код формы в соответствии с примером.
3. Отредактируем файл стилей, чтобы определить стили для новой формы.
4. Запустим и протестируем приложение

**Лабораторная работа 3. Применение операторов C# в коде страницы**

**Упражнение 2. Вызов метода из отделенного кода**

1. Перенесем логику заполнения таблицы в отдельный файл. Для этого определим метод GetNoShowHtml в классе Summary.
2. Отредактируем файл Summary.aspx, вставив результат выполнения данного метода на страницу.
3. Протестируем работу и удостоверимся, что поведение приложения осталось неизменным.

**Лабораторная работа 4. Использование Master Page при построении интернет-приложений**

**Упражнение 1. Добавление и применение главной страницы**

1. Добавим в проект главную страницу и добавим на нее таблицу, выступающую в качестве сетки отображения контента.
2. Добавим элемент ContentPlaceHolder, на месте которого будет отображаться подставляемое заполнение.
3. Отредактируем параметры страницы, добавив заголовок и стили.

**Лабораторная работа 4. Использование Master Page при построении интернет-приложений**

**Упражнение 2. Преобразование веб-форм в страницы содержимого**

1. Преобразуем ранее созданные страницы, оставив только содержимое тега body, для их дальнейшего встраивания в поле заполнения главной страницы.
2. Добавим на главную страницу возможность навигации, определив элементы меню в сайдбаре и футере.
3. Запустим и протестируем приложение, чтобы удостовериться в корректной работе.

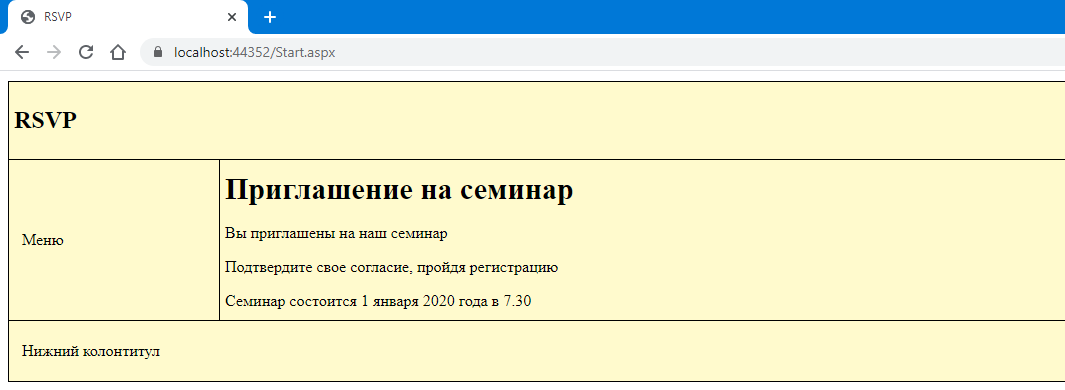


Рисунок 1 - Стартовая страница приложения.

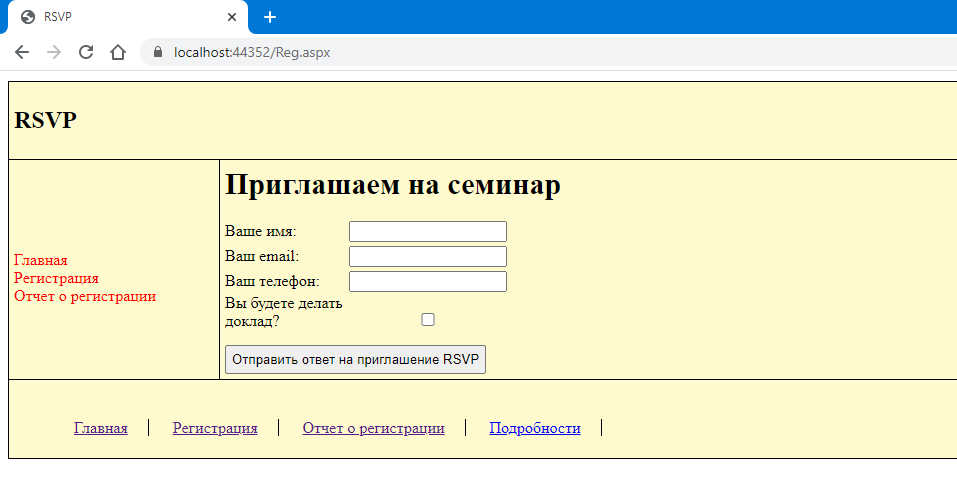


Рисунок 2 - страница регистрации на семинар.

1. Добавим веб-форму About.aspx как страницу содержимого и настроим ее внешний вид.
2. Проверим корректность отображения.

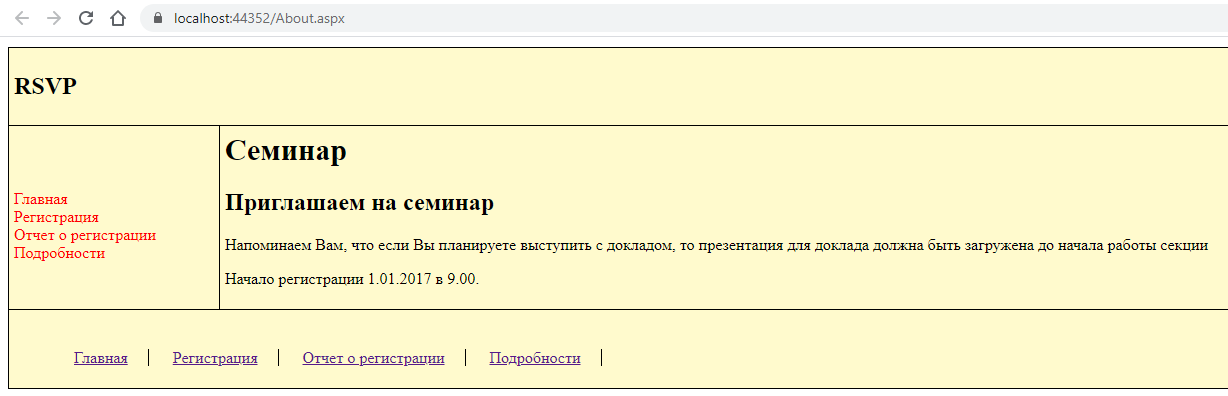


Рисунок 3 - проверка страницы About

**Лабораторная работа 5. Выполнение проверки достоверности**

**Упражнение 1. Добавление проверяющих элементов управления**

1. Отредактируем форму регистрации добавив компоненты валидации **asp:RequiredFieldValidator** для полей формы. Таким образом заполняемые поля являются обязательными и вызывают ошибку с предупреждением при попытке отправить форму без их заполнения, а также вводе некорректных данных, которые не соответствуют определенному шаблону.
2. Протестируем работу приложения.

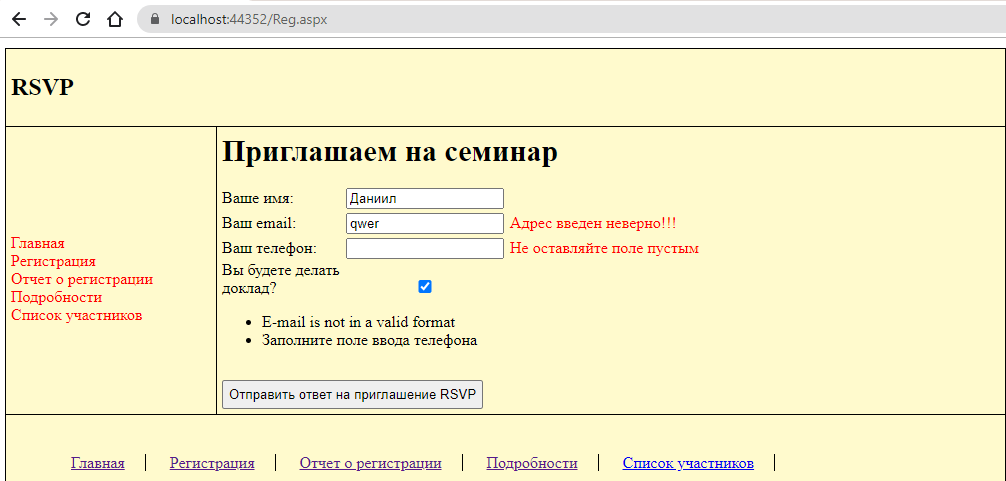


Рисунок 4 - тестирование, валидация полей формы.

**Лабораторная работа 6. Применение технологии ASP.NET AJAX**

**Упражнение 1. Анимирование элемента управления UpdatePanel**

1. Добавим на страницу таймер с отчетом времени до начала семинара с интервалом обновления 1 секунда.
2. Протестируем работу приложения.

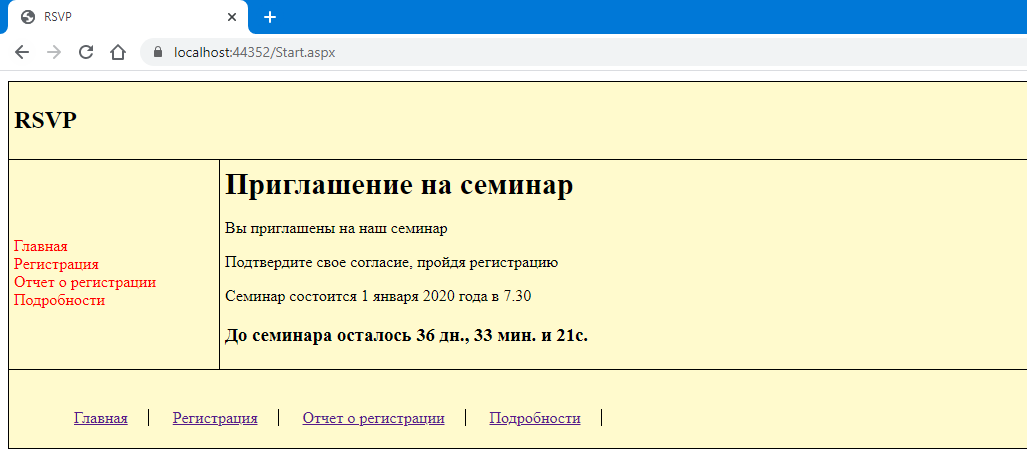


Рисунок 5 - добавление обратного отсчета до начала семинара.

**Лабораторная работа 7. Управление состоянием в веб-приложениях**

**Упражнение 1. Отображение значения счетчика посещений**

1. Добавим счетчик посещения нашего сайта. Для этого создадим глобальный класс Global.asax и определим переменную с именем Visitors.
2. В нижней секции главной страницы добавим вывод информации о количестве пользователей, посетивших наш сайт. Для этого в футере определим место для вывода информации с помощью тега asp:Literal, а в обработчике события Page\_Load выполним инициализацию и присвоение информации о числе посещений.
3. Протестируем приложение.

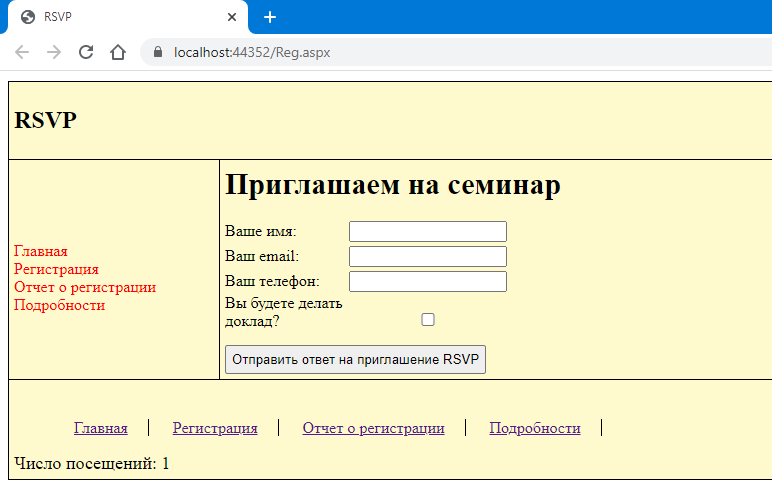


Рисунок 6 - тестирование счетчика посещений

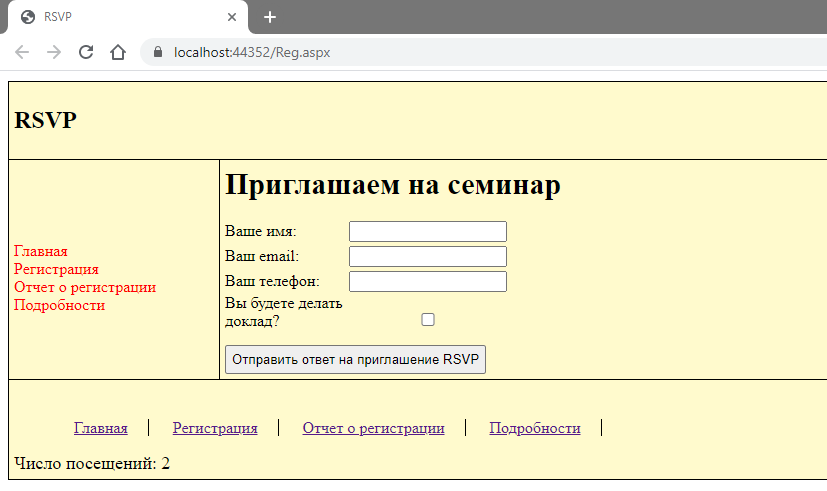
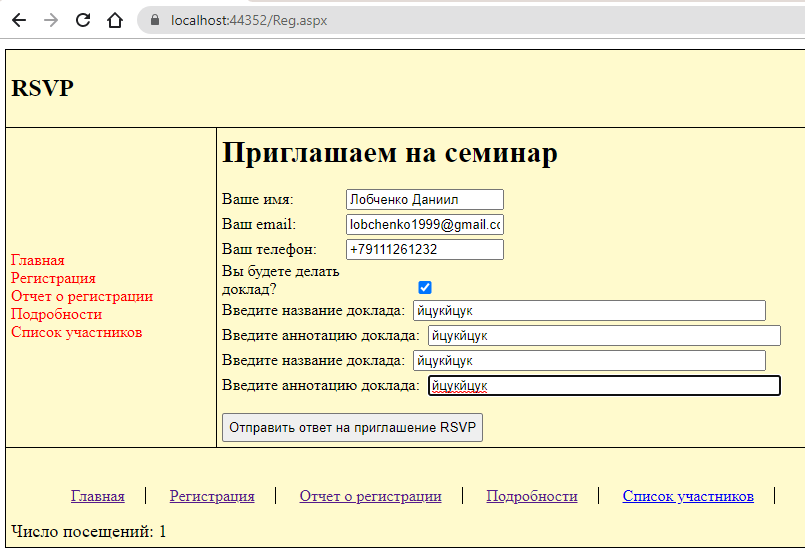


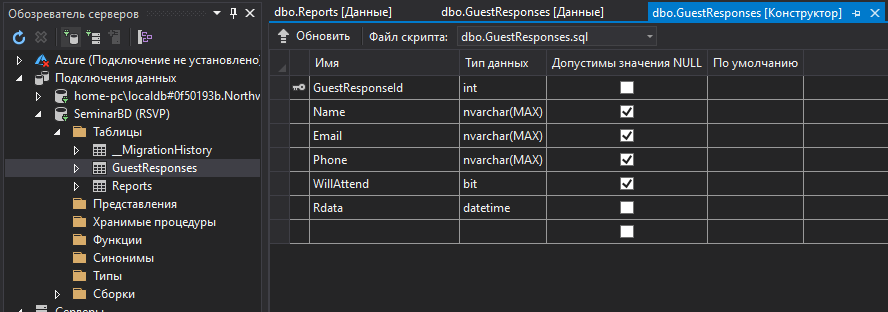
Рисунок 7 - тестирование счетчика посещений

**Лабораторная работа 8. Работа с базой данных**

**Упражнение 1. Использование Code-First**

1. Откроем и отредактируем файл GuestResponse.cs, добавив новый класс Report, моделирующий сущность «доклад».
2. В классе GuestResponse добавим ссылку на коллекцию докладов (Reports).
3. Изменим интерфейс формы, добавив возможность выбрать число докладов и ввести информацию о каждом из докладов.
4. Установим Entity Framework 6 в проект
5. Для создания нового класса контекста добавим в проект новый класс SampleContext.
6. Определим имя базы данных SeminarDB и модели нашей базы данных DbSet<GuestResponse> и DbSet<Report>.
7. В обработчике загрузки формы Reg.aspx.cs определим создание участника и докладов, а также добавление их в базу данных.
8. Протестируем приложение.

Рисунок 8 - Заполнение форма регистрации на семинар

Рисунок 9 - проверка создания базы данных

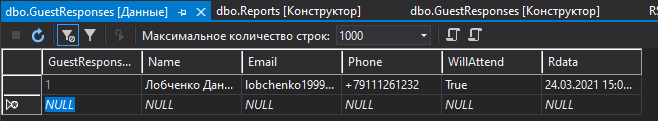


Рисунок 10 - Проверка записи информации в базу данных

**Лабораторная работа 8. Работа с базой данных**

**Упражнение 2. Отображение данных**

1. Добавим в проект сайта новую форму SummaryBD.aspx, как страницу содержимого.
2. Добавим на форму компоненты GridView и LinqDataSource и настроим источник данных компонента LinqDataSource.

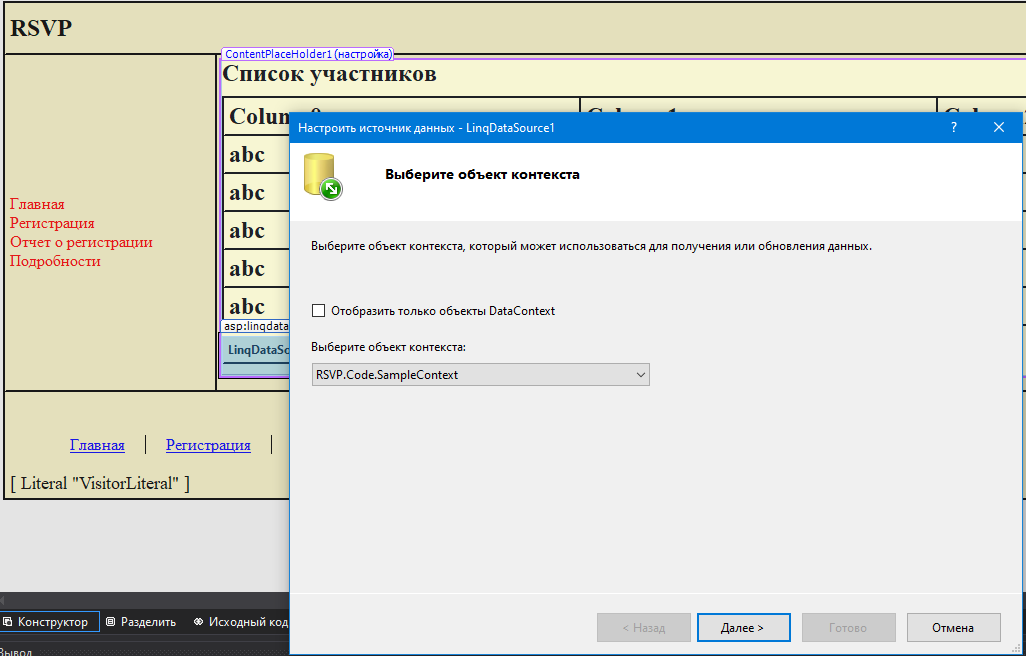


Рисунок 11 - настройка источника данных компонента LinqDataSource

1. Настроим источник данных для элемента GridView.
2. Настроим отображение столбцов для элемента GridView.

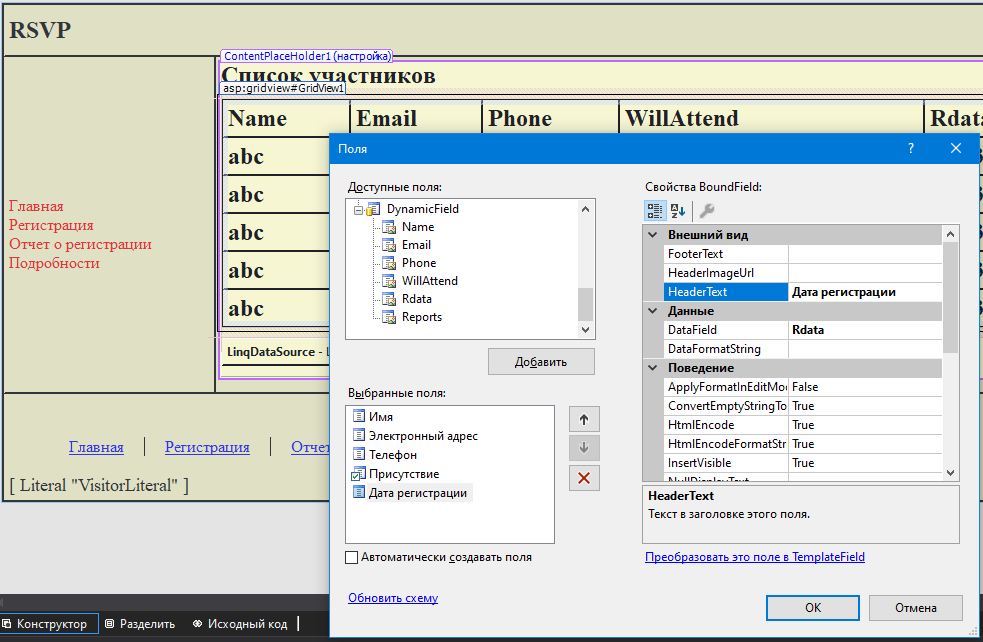


Рисунок 12 - настройка отображения столбцов GridView

1. Запустим и протестируем приложение.

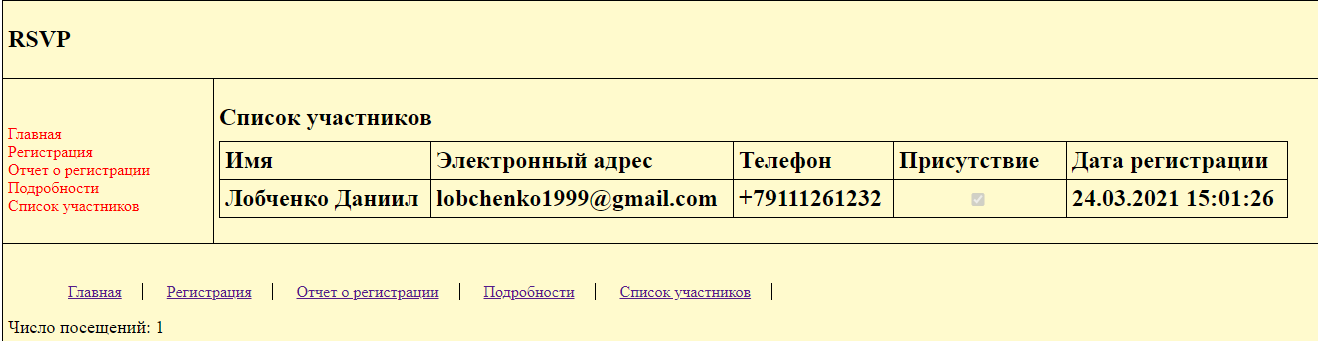


Рисунок 13 - тестирование приложения

**Вывод**

В данной лабораторной работе было разработано веб-приложение с использованием технологии ASP.NET. По выполнении лабораторной работы можно отметить, что данная технология является довольно-таки мощным инструментом для построения полноценных веб-приложений. Поддержка рендеринга исполнения и встраивания с# кода в шаблоны страниц является большим плюсом, что сокращает время до первого отклика страницы.

Шаблонная система позволяет переиспользовать существующий код при проектировании страниц приложения, применяя шаблоны страниц с элементами, позволяющими подставить необходимый контент.

Использование данной технологии позволяет использовать средства экосистемы .NET при создании браузерных приложений, что сильно упрощает работу с базами данных и другими компонентами. Данная технология поддерживает обновление страниц в режиме реального времени, что позволяет реализовывать счетчики, таймеры и прочие элементы интерфейса, логика которых завязана на серверных элементах системы.