**Лабораторная работа №1 «Создание основы веб-сайта приглашения на**

**семинар»**

**Постановка задачи**

Требуется создать веб-сайт, который дал бы возможность приглашать на семинар, а приглашенным отправлять ответы на приглашение (repondez s'il vous plait - RSVP).

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

В Visual Studio создали Web-Site с шаблоном ASP.NET Empty Web Site. Шаблон ASP.NET Empty Web Site – простейший из всех шаблонов сайтов. Он создает сайт, содержащий лишь файл Web.config, в котором находится конфигурационная информация для приложения ASP.NET.

Для дальнейшей работы с данными создали модель данных, добавив в проект класс GuestResponse с автоматически реализуемыми свойствами. Также для данных нужно хранилище. В дальнейшем это будет база данных, но на первом этапе реализовали хранение объектов в памяти, для чего создали класс ResponseRepository. Преимущество такого подхода связано с простотой реализации, однако следует учитывать, что данные будут теряться при каждом останове или перезапуске приложения, поэтому лучше сохранять коллекцию в файл с помощью механизма сериализации.

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

Добавили в проект новую стартовую форму Start.aspx, которая выводит несколько информационных предложений на экран.

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 1.1 – Стартовая форма простого сайта*

Далее добавили еще одну форму для регистрации Reg.aspx. Данная форма содержит серверные элементы, предназначенные для сбора информации от пользователей и кнопку для ее отправки.

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 1.2 – Форма регистрации*

Для того чтобы придать элементам формы определенный вид к ним нужно применить CSS стиль. Элементы веб-формы стилизуются с применением каскадных таблиц стилей (Cascading Style Sheets - CSS). Добавили в проект новый файл Styles.css, после чего добавили элемент link (таблица стилей CSS ассоциируется с веб-формой с помощью этого элемента) в раздел head файлов Reg.aspx и Start.aspx. В результате написанные стили применились к формам.

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 1.3 – Форма регистрации со стилем*

**Лабораторная работа №2 «Обработка событий формы»**

В данной работе вы реализовали реакцию на отправку данных на семинар как реакцию на кнопку "Отправить ответ приглашение RSVP".

**Упражнение 1. Обработка события загрузки формы**

Базовым для класса отделенного кода является класс System.Web.UI.Page, который содержит несколько методов и свойств для ответа на веб-запросы. По умолчанию добавлен пустой метод Page\_Load(), вызываемый ASP.NET при поступлении запросов для Reg.aspx и предоставляющий возможность отреагировать на такие запросы. В данном случае метод Page\_Load() будет вызван один раз во время первоначальной загрузки страницы и еще раз, когда пользователь отправит форму.

В обработчике события загрузки формы сначала с помощью проверки свойства IsPostBack определили, относится ли запрос, на который производится ответ, к форме, отправленной обратно серверу, и если это так, то создаем новый экземпляр объекта GuestResponse модели данных с полученными от элементов формы данными. Затем объект GuestResponse помещаем в хранилище.

Пользователю необходимо предоставить какой-либо отклик после того, как он отправил форму, и это делается с помощью метода Response.Redirect(), который выполняет перенаправление пользовательского браузера. Для этого реализовали проверку свойства WillAttend: если оно равно true, пользователь будет делать на семинаре доклад, поэтому он перенаправляется на файл seeyouthere.html. В противном случае перенаправление происходит на файл sorryyoucantcome.html.

**Упражнение 2. Создание HTML-файлов**

Создали 2 новые HTML-страницы с контентом для отображения пользователю после регистрации на семинар. В зависимости от того, будет ли пользователь делать доклад на семинаре он будет перенаправлен на первую или вторую страницу соответственно.

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 2.1 – Заполнение формы регистрации с докладом*

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 2.2 – Страница пользователю, выступающему с докладом*

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 2.3 – Заполнение формы регистрации без доклада*

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 2.4 – Страница пользователю, не выступающему с докладом*

**Лабораторная работа №3 «Применение операторов C# в коде страницы»**

**Упражнение 1. Создание итогового представления**

Добавили в проект еще одну форму Summary.aspx, которая отображает итоговые сведения по полученным ответам, чтобы можно было видеть, кто планирует выступить на семинаре с докладом. В теле формы содержится заголовок таблицы и, внутри блока <tbody>, код C#, который обеспечивает заполнение таблицы с помощью запроса LINQ. В нём дополнительно применяются дескрипторы <% и %> для добавления динамического контента к генерируемому HTML-коду, когда браузер запрашивает данный файл.

Внутри блока кода используются обычные операторы C# для генерации набора HTML-элементов, представляющих собой строки в элементе table, в этих строках перечислены люди, принявшие решение выступать на семинаре с докладом. Для получения всех объектов данных из хранилища вызывается метод ResponseRepository.GetRepository().GetAllResponses(), а для выборки всех подтверждающих участие ответов применяется LINQ-выражение Where(). Затем в цикле foreach выполняется запрос и генерируются HTML-строки для каждого объекта данных. Метод String.Format() позволяет компоновать HTML-строки, содержащие значения свойств из каждого объекта GuestResponse, который необходимо отобразить. Для добавления HTML-кода в вывод, отправляемый браузеру, используется метод Response.Write(). Также В файл Summary.aspx был включен элемент link, который импортирует файл Styles.css с содержащимися внутри него стилями.

Для проверки работы системы проведем несколько отправок заявок на семинар и проверим результат.

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 3.1 – Заполнение формы Студент 1*

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 3.2 – Заполнение формы Студент 2*

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 3.3 – Заполнение формы Студент 3*

Изображение выглядит как снимок экрана

Автоматически созданное описание

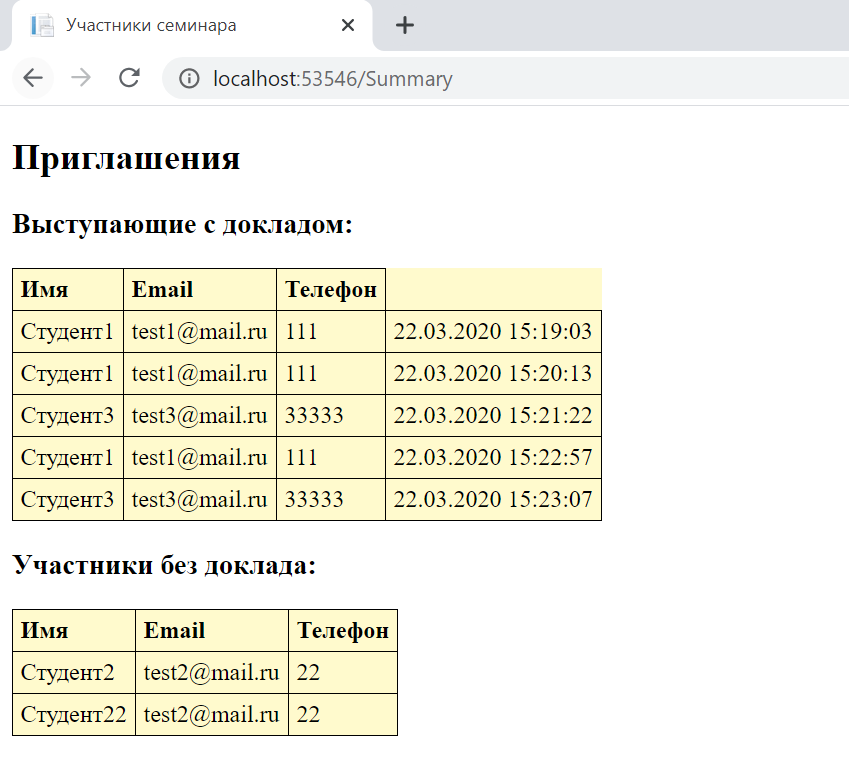
*Рисунок 3.4 – Таблица участников с докладом*

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

В этом упражнении применили другой подход, который предусматривает определение методов в файле отделенного кода, а затем использование фрагментов кода для вызова этих методов и вставки результатов в HTML-код, отправляемый браузеру.

В файле отделенного кода Summary.aspx.cs определите новый метод GetNoShowHtml(), который генерирует таблицу строк с теми, кто не будет выступать с докладом. В режиме разметки формы Summary.aspx добавили разметку для формирования таблицы и вызвали новый метод внутри тега <tbody>.

В этом примере применяется фрагмент кода с открывающим дескриптором <%=. Это сообщает ASP.NET о необходимости вставки результата выполнения метода в вывод, отправляемый браузеру, что представляет собой более аккуратный и читабельный подход, чем включение кода непосредственно в страницу. Генерируется такой же HTML-код, что и с помощью предыдущего фрагмента кода, но только в данном случае получается таблица строк для людей, которые не выступают с докладом.



*Рисунок 3.5 – Все участники семинара*

**Лабораторная работа №4 «Использование Master Page при построении интернет-приложений»**

В этой работе обеспечили единообразный пользовательский интерфейс для всех клиентов. Для реализации такой возможности добавили главную (мастер) страницу (Master Page) и преобразовали существующие веб-формы в страницы содержимого. Мастер-страница содержит фиксированные элементы, одинаковые для всех страниц, и заполнитель содержимого для остальной части страницы. Наиболее типичными фиксированными элементами являются верхний и нижний колонтитулы, панель навигации, панель меню и т.д. Страница содержимого получает от мастер-страницы фиксированные элементы и предоставляет дополнительное содержимое.

**Упражнение 1. Создание итогового представления**

В этом упражнении добавили в существующий веб-сайт главную страницу, разработав ее структуру (верхний, нижний колонтитулы и панель навигации – общие для всех страниц элементы) и определили элемент управления ContentPlaceHolder.

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 4.1 – Конструктор главной страницы*

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

В этом упражнении преобразовали веб-форму по умолчанию в страницу содержимого и добавили в главную страницу возможности навигации.

В директиву Page добавили свойство MasterPageFile и установите его значение равным ~/Site.master. После директивы Page добавили серверный элемент управления asp:Content, в который поместили основной контент страницы. Применив данную операцию ко всем созданным страницам, проверили, что стиль главной страницы применился к каждой из них.

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 4.2 – Страница Start со стилем главной страницы*

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 4.3 – Страница Reg со стилем главной страницы*

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 4.4 – Страница Summary со стилем главной страницы*

Далее добавили в область меню навигацию по страницам с помощью элемента управления MainMenu, завернутый в div-элемент со значением атрибута класса menu, ориентацией вертикально и цветом команд красный.

Также в нижний колонтитул добавили ссылки на страницы в виде маркированного списка, устанавливаемый тегом <ul>.

В файле со стилями Styles.css прописали параметры для форматирования списка ссылок.

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 4.5 – Стартовая страница с меню навигации*

**Упражнение 3. Добавление веб-формы как страницы содержимого**

В этом упражнении добавили веб-форму About.aspx как страницу содержимого, указав при добавлении главную страницу – Site.master. В код страницы добавили описание семинара. Также добавили в файл главной страницы в коллекцию меню новый элемент с указанием пути к новой странице.

Изображение выглядит как снимок экрана

Автоматически созданное описание

*Рисунок 4.6 – Страница «Подробности»*

Лабораторная работа 5. Выполнение проверки достоверности

Упражнение 1. Добавление и применение главной страницы

Была реализована проверка вводимых пользователем данных при регистрации. Для этого использовались элементы управления RequiredFieldValidator, который проверяет были ли введены данные в указанное поле и выводит сообщение ошибки, в случае если данные не были введены и ValidationSummary, который отображает все сообщения об ошибках (Рисунок ).

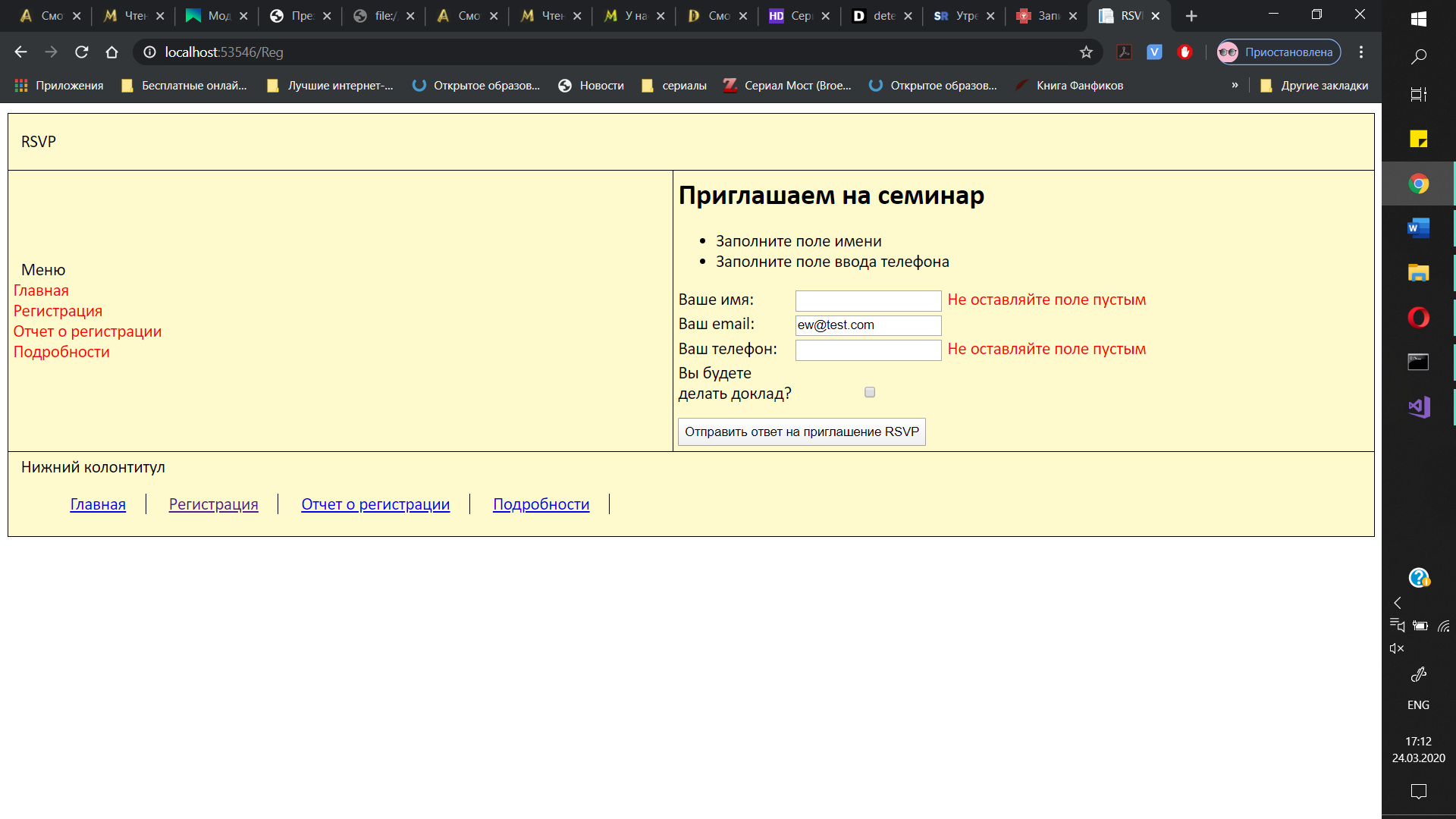


Рисунок 1 – проверка достоверности введённых данных

Лабораторная работа 6. Применение технологии ASP.NET AJAX

Упражнение 1. Анимирование элемента управления UpdatePanel

На стратовую страницу был добавлен анимированный элемент: счетчик до даты семинара. Для его созания были использованы элементы AJAX: ScriptManager и UpdatePanel. В элементе ContentTemplate элемента UpdatePanel был добавлен код вычисляющий разность дней, минут и секунд между назначенной датой семинара и текущей датой (Рисунок ).

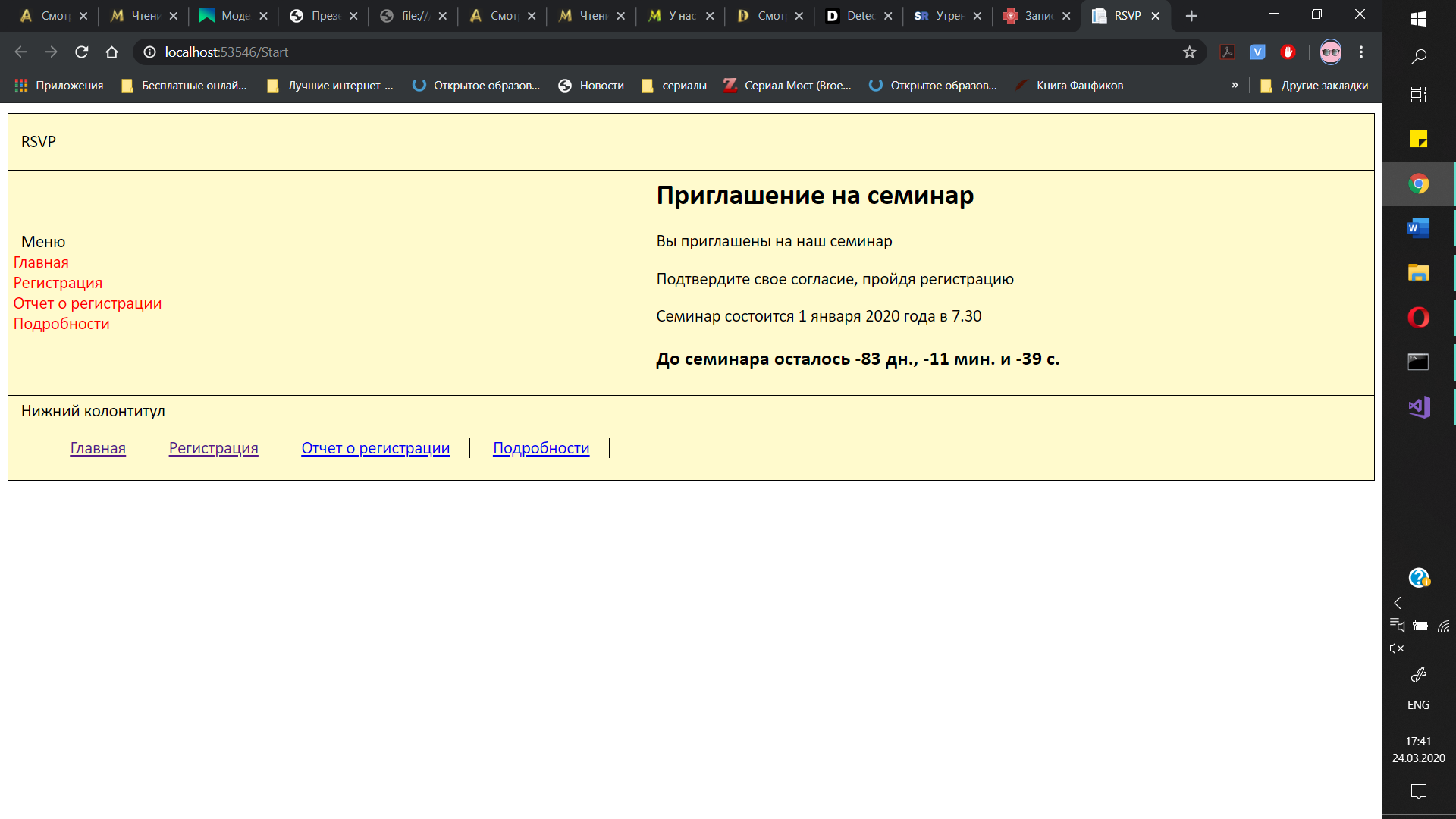


Рисунок 2 –

Лабораторная работа 7. Управление состоянием в веб-приложениях

Упражнение 1. Отображение значения счетчика посещений

Был добавлен счетчик посещений. Для этого на главной странице был добавлен элемент Literal. В классе Global.asax был добавлена переменная для подсчета количества посетителей и было реализовано изменение значение переменной при начале новой сессии. Также локальная переменная была добавлена в код главной формы, она берет значение из класса Global и выводит в элемент Literal (Рисунок ).

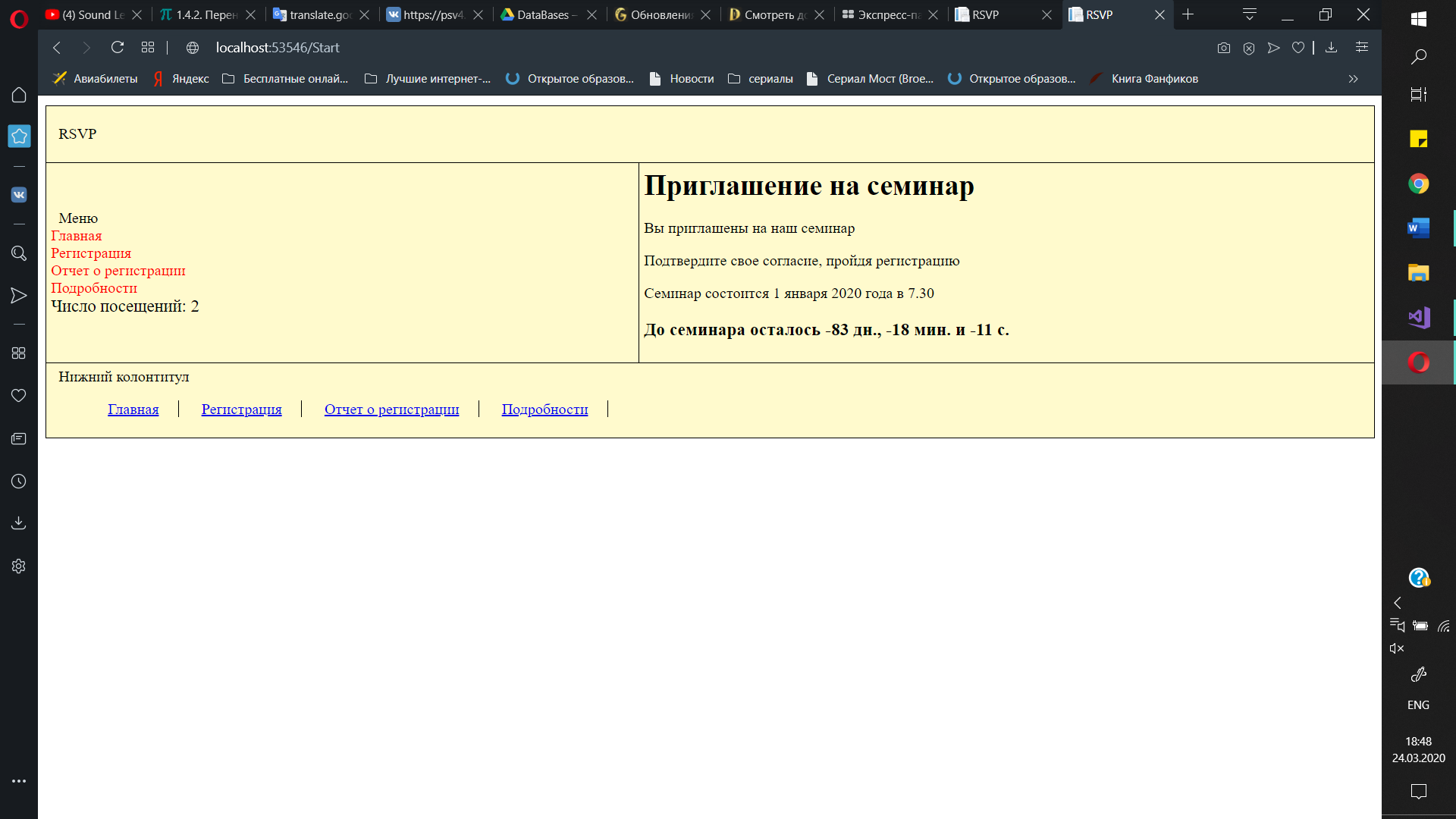


Рисунок –

Лабораторная работа 8. Работа с базой данных

Упражнение 1. Использование Code-First

Была изменена модель данных: в файл GuestResponse.cs был добавлен новый класс Report, моделирующий сущность «доклад». Было установлено подключение к базе данных SeminarBD.