

Министерство образования и науки РФ
Казанский национальный исследовательский технический университет имени
А. Н. Туполева
Факультет технической кибернетики и информатики
Кафедра систем информационной безопасности

Создание фотогалереи

Отчет по лабораторной работе

Выполнили:

студенты группы 4305

Галимов Марат, Мухамбеталиев Дильшат

Казань 2013

Содержание

1 Введение

HTML (HyperText Markup Language, язык разметки гипертекста) — это система верстки, которая определяет, как и какие элементы должны располагаться на веб-странице. Информация на сайте, способ ее представления и оформления зависят исключительно от разработчика и тех целей, которые он перед собой ставит.

Стилем или CSS (Cascading Style Sheets, каскадные таблицы стилей) называется набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид. Возможность работы со стилями издавна включают в развитые издательские системы и текстовые редакторы, тем самым позволяя одним нажатием кнопки придать тексту заданный, заранее установленный вид. Теперь это доступно и создателям сайта, когда цвет, размеры текста и другие параметры хранятся в определенном месте и легко «прикручиваются» к любому тегу. Еще одним преимуществом стилей является то, что они предлагают намного больше возможностей для форматирования, чем обычный HTML.

Хостинг (англ. hosting) — услуга по предоставлению вычислительных мощностей для размещения информации на сервере, постоянно находящемся в сети (обычно Интернет). Хостингом также называется услуга по размещению оборудования клиента на территории провайдера с обеспечением подключения его к каналам связи с высокой пропускной способностью (колокация, от англ. collocation).

База данных (БД) - это информационная модель, позволяющая в упорядоченном виде хранить данные о группе объектов, обладающих одинаковым набором свойств.

NoSQL (англ. not only SQL, не только SQL), в информатике — термин, обозначающий ряд подходов, проектов, направленных на реализацию моделей баз данных, имеющих существенные отличия от используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL. Описание схемы данных в случае использования NoSQL-решений может осуществляться через использование различных структур данных: хеш-таблиц, деревьев и других.

Цели и методы: Целью созданной страницы является получение рабочей интернет-страницы.

Практическое применение: страницу можно разместить в интернете и сделать ее полностью функционирующей.

Используемые технологии:

Хостинг Windows Azure.

Windows Azure — это открытая и гибкая облачная платформа, которая позволяет вам быстро выполнять построение приложений, развертывать их и управлять ими в рамках глобальной сети из центров данных, управляемых корпорацией Майкрософт. Вы можете осуществлять построение приложений с помощью любого языка, средства или любой платформы. Вы также можете интегрировать свои общедоступные облачные приложения с существующей ИТ-средой.

адрес: `gmmd.azurewebsites.net`

MongoDB — концептуально то же самое, что обычная, привычная нам база данных (или в терминологии Oracle — схема). Внутри MongoDB может быть ноль или более баз данных, каждая из которых является контейнером для прочих сущностей.

База данных может иметь ноль или более «коллекций». Коллекция настолько похожа на традиционную «таблицу», что можно смело считать их одним и тем же.

Коллекции состоят из нуля или более «документов». Опять же, документ можно рассматривать как «строку».

Документ состоит из одного или более «полей», которые — как можно догадаться — подобны «колонкам».

«Индексы» в MongoDB почти идентичны таковым в реляционных базах данных.

«Курсоры» отличаются от предыдущих пяти концепций, но они очень важны (хотя порой их обходят вниманием) и заслуживают отдельного обсуждения. Важно понимать, что когда мы запрашиваем у MongoDB какие-либо данные, то она возвращает курсор, с которыми мы можем делать все что угодно — подсчитывать, пропускать определённое число предшествующих записей — при этом не загружая сами данные.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using MongoDB.Bson;
using MongoDB.Driver;
using MongoDB.Driver.Builders;

namespace PG
{
    public class DataAccess
    {
        private MongoCollection<Foto> GetCollection()
        {
            string connectionString =
                "mongodb://qwerty:qwerty@ds053828.mongolab.com:53828/lr1";
            var client = new MongoClient(connectionString);
            var server = client.GetServer();
            var db = server.GetDatabase("lr1");
            var collection = db.GetCollection<Foto>("db1");
            return collection;
        }

        public void Insert(Foto foto)
        {
            var collection = GetCollection();
            collection.Insert(foto);
        }

        public List<Foto> GetFoto1()
        {
            var collection = GetCollection();
            return collection.FindAllAs<Foto>().ToList();
        }

        public void Remove(int myId)
        {
            var collection = GetCollection();
            var query = Query<Foto>.EQ(e => e.MyID, myId);

```

```
        collection.Remove(query);  
    }  
}  
}
```

Есть четыре основных типа запросов данных в NoSQL, которые относятся к так называемому языку манипулирования данными: SELECT – выбрать строки из таблиц; INSERT – добавить строки в таблицу; UPDATE – изменить строки в таблице; DELETE – удалить строки в таблице;

но мы использовали только два: Запрос INSERT используется для создания новой строки данных. Для обновления уже существующих данных или пустых полей строки нужно использовать запрос UPDATE. синтаксис запроса INSERT:

```
INSERT INTO table_name (column1, column2, column3)
VALUES ('data1', 'data2', 'data3');
```

Запрос DELETE полностью удаляет строку из базы данных. Если вы хотите удалить одно единственное поле, то нужно использовать запрос UPDATE и установить для этого поля значение, которое будет являться аналогом NULL в вашей программе. Будьте внимательны, и ограничивайте ваш запрос DELETE условием WHERE, иначе вы можете потерять все содержимое таблицы.

```
DELETE FROM table_name WHERE column1 = 'data1';
```

пользовательский интерфейс заключается в виде одной страницы на которой находятся кнопки обзор и добавить(рис. 1), при добавлении фотографии над ней автоматически создается кнопка удалить(рис 2.).

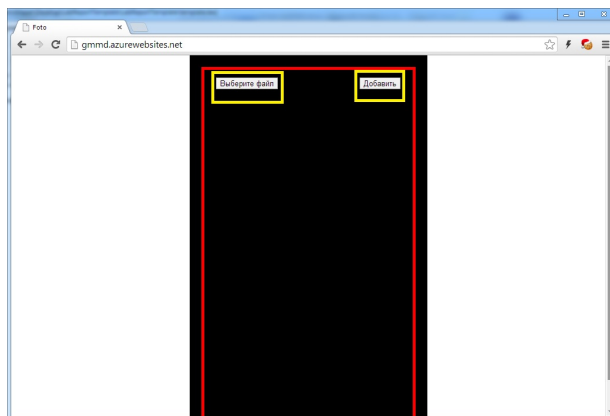


рис.1)

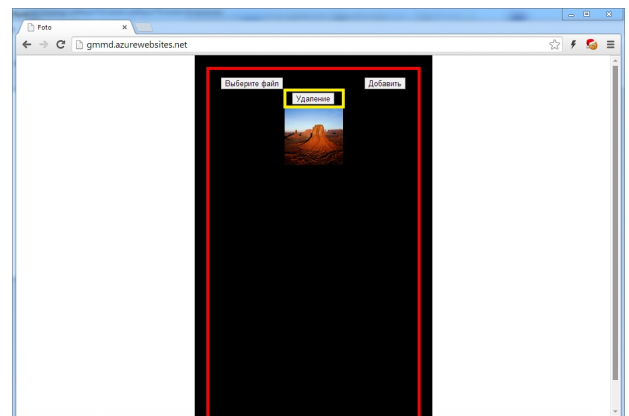


рис.2)

2 Ход работы и описание

1)Постановка задачи

Перед тем как приступить к разработке сайта, мы определили цели и задачи проектируемого ресурса. В зависимости от этого определили структуру и тип сайта.

2)Верстка

Верстка - процесс создания статических страниц сайта с текстом и графикой таким образом, чтобы это выглядело так, как задумывалось дизайнером. Результатом верстки обычно является один или несколько файлов (.html) - статических страниц сайта, картинки (.png, .jpg, .gif) - участвуют в оформлении сайта, скрипты (.js) - для реализации дополнительных визуальных возможностей сайта, стили (.css) - каскадные таблицы стилей.


```

body {
    margin: 0 30% ;

}
.parent
{

    background: #000000;
    padding: 10px;

}
.child {
    border: 5px solid red;
    padding: 10px;
    margin: 10px;

}

```

```

#menu_it a {

    text-decoration: none;

    color: #003388;
    background-color: #f5f5f5;
    margin: 0 5px 0 0;
    border-radius: 5px;

    display: inline-block;
    padding: 8px 18px;

}

```

```

#menu_it a:hover {
    color: #333;
    background-color: red;
    text-decoration: none;
}

```

```
img
{
    display: inline-block;
    margin: 1%;
}
```

3) Программирование.

Эта часть работы заключается в написании для сайта программ, необходимых для его реализации.

показан код работы кнопок

```

using System;
using System.Collections.Generic;
using System.Web.UI.WebControls;
namespace PG
{
    public partial class Default : System.Web.UI.Page
    {
        public List<Foto> Foto1
        {
            get { return new DataAccess().GetFoto1(); }
        }

        protected void Page_PreRender(object sender, EventArgs e)
        {
            DataBind();
        }

        //remove button
        protected void btn_Click(object sender, EventArgs e)
        {
            var button = sender as Button;
            var id = Convert.ToInt32(button.CommandArgument);

            new DataAccess().Remove(id);
        }

        protected void Add_Click(object sender, EventArgs e)
        {
            if (OurFileUpload.HasFile)
            {
                string path = Server.MapPath("//");
                OurFileUpload.SaveAs(path + OurFileUpload.FileName);

                var currentFoto = new Foto();
                currentFoto.jpg = OurFileUpload.FileName;
                new DataAccess().Insert(currentFoto);
            }
        }
    }
}

```

4)Тестирование

Подробная проверка работы и качества сайта в различных условиях, функционирование в различных типах браузеров (IE6,IE7,IE8,FF,Opera,Chrome,Safari)

5)Публикация в сети Интернет

Непосредственное размещение сайта в сети Интернет gmmd.azurewebsites.net

3 Заключение

Целью данного задания было составить рабочую фотогалерею с элементами добавления и удаления фотографий.

Был составлен соответствующий сайт, с рабочими элементами .