

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

ПО ДИСЦИПЛИНЕ

«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ»

НА ТЕМУ

«Создание приложения ASP.NET с хранением данных в формате XML»

Вариант – 2

Факультет: АВТ Преподаватель: Дубков И.С.

Группа: АВТ-918

Студент: Ванин К.Е.

Цель работы

Освоение методики создания приложений ASP.NET с хранением данных в формате XML.

Задание

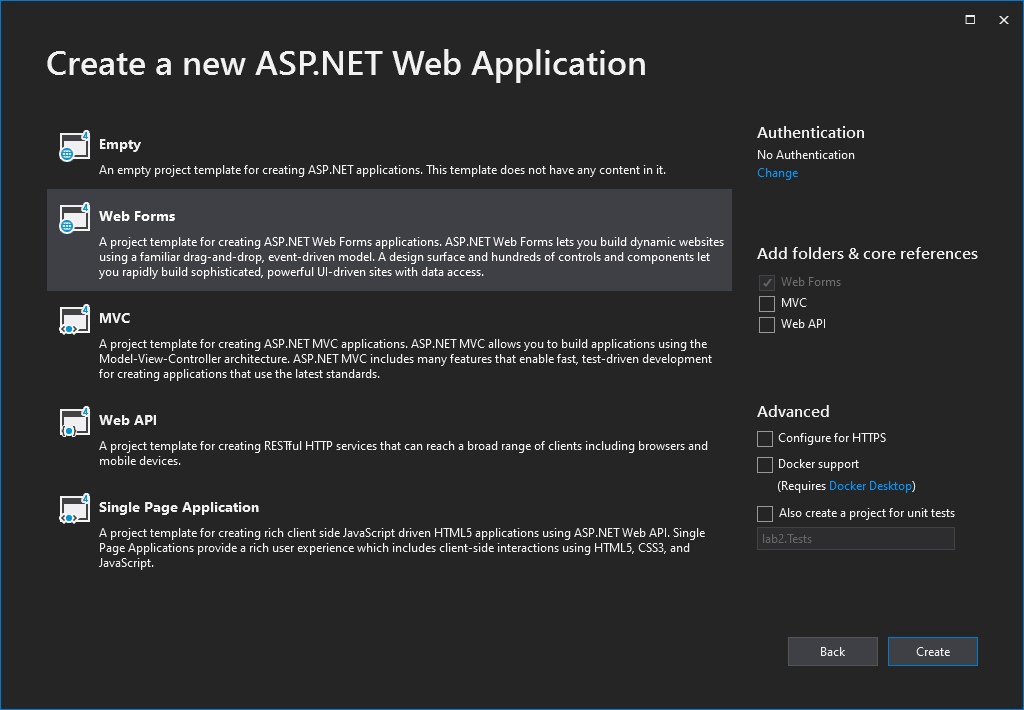
1. Создать Web-приложение ASP.NET и с его помощью создать резидентную базу данных по технологии ADO.NET в соответствии с вариантом задания.
2. Заполнить созданную базу данными, введенными из Веб-форм.
3. Просмотреть содержимое базы данных в визуальном элементе веб-формы.
4. Выполнить модификацию базы данных.
5. Сохранить данные в файле XML.
6. Выполнить очистку базы данных.
7. Реализовать загрузку данных из файла XML в резидентную базу данных.
8. Реализовать запрос к базе данных на получение требуемой информации.
9. Результаты выполнения пунктов задания отобразить в отчете.

**Вариант** 2:

Коллекция картин художественного музея

Ход работы

Для выполнения лабораторной работы был выбран ASP.NET Web Forms для системы Windows.

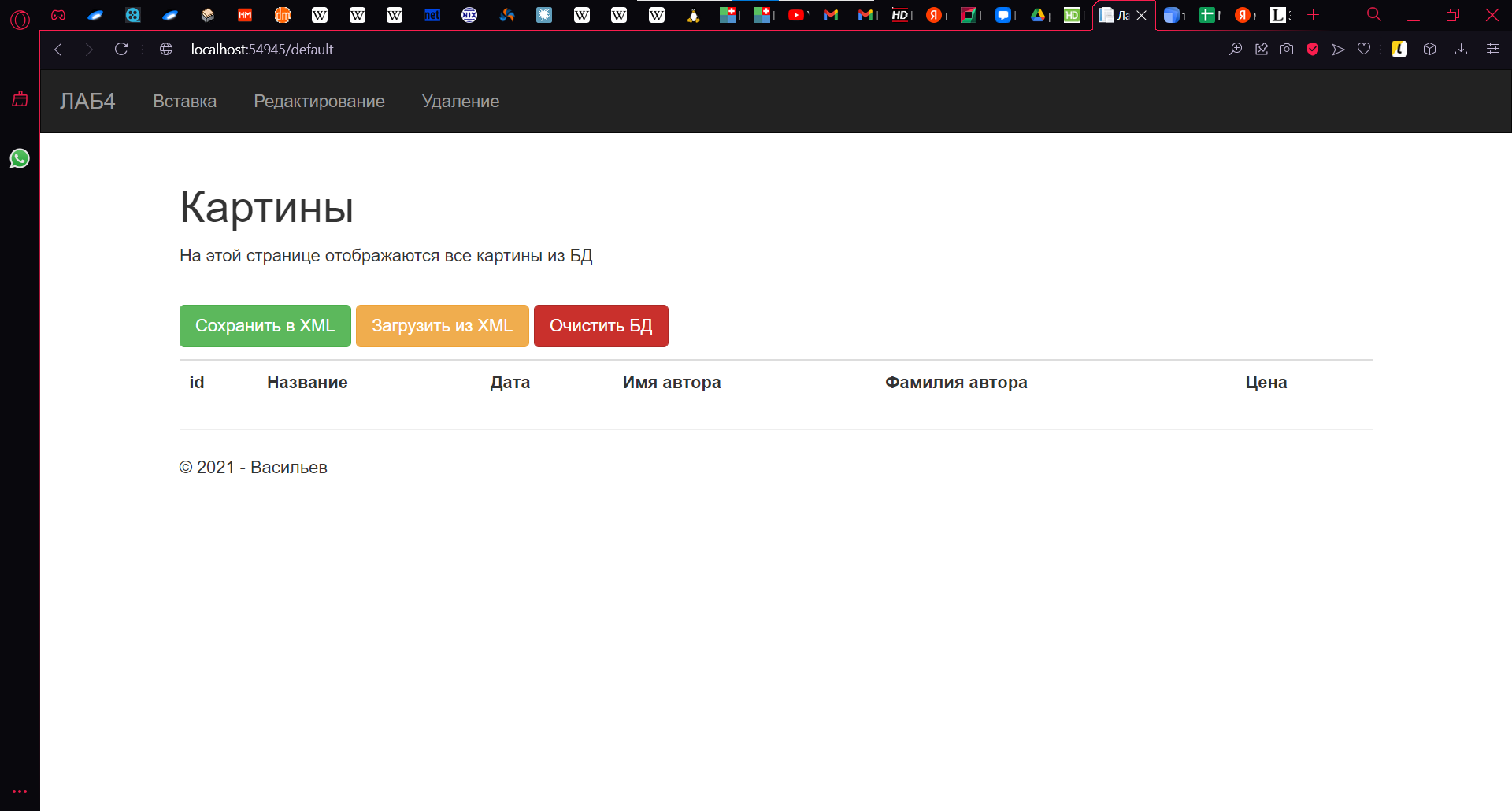


*Рис. – Создание проекта*

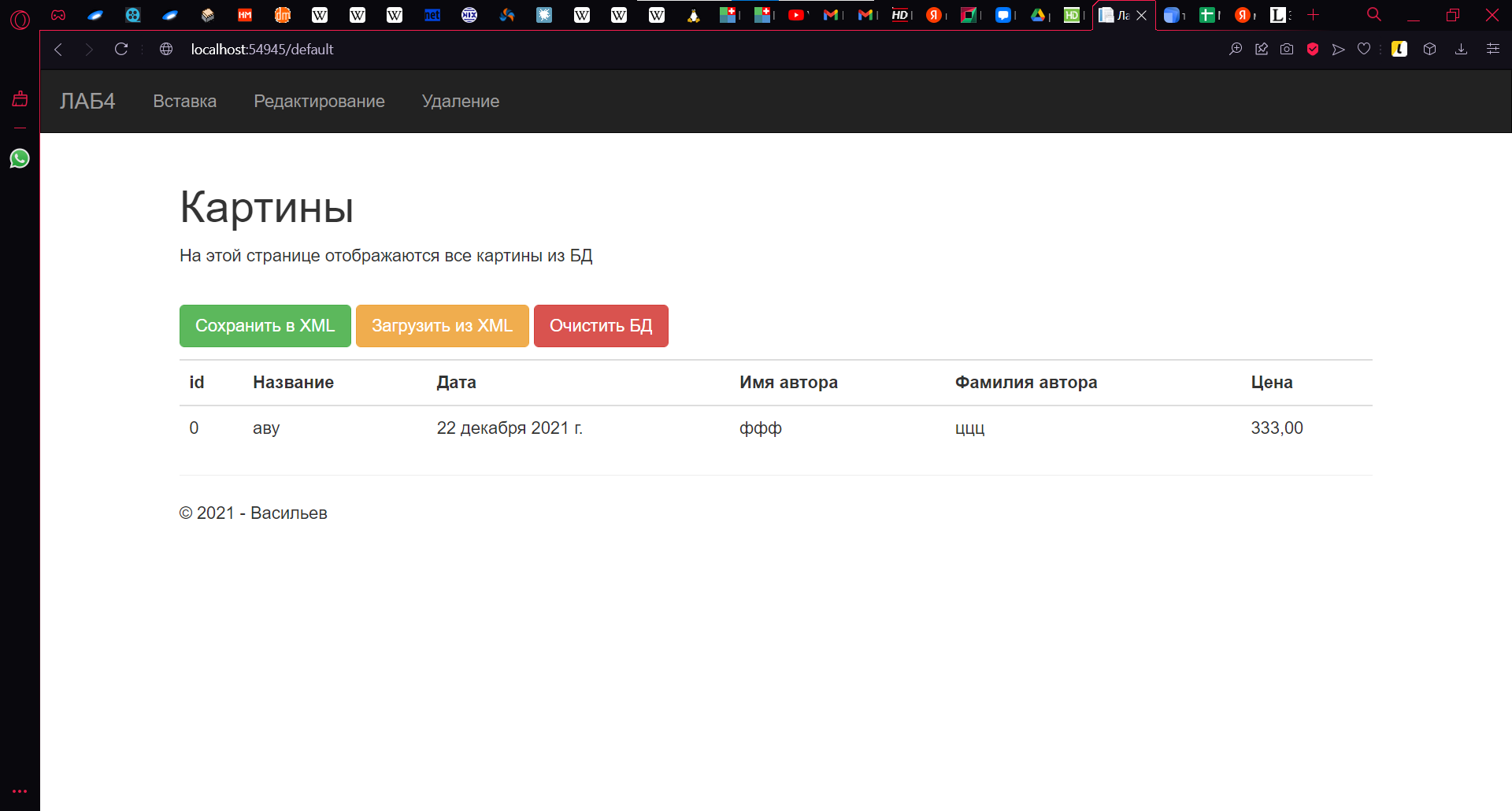
Согласно заданию, был создан пустой проект, в который была добавлена мастер-страница с именем по умолчанию (Site.Master). Созданная мастер-страница представляет собой «каркас» сайта – она содержат HTML-разметку с шапкой и подвалом сайта (также туда входит все, что не относится к контентной части).

В Global.asax добавили собственный класс, который хранит в оперативной памяти резидентную базу данных, которая позволяет нам работать с ней, даже при переходе по страницам сайта. При запуске приложения мы создаем DataSet, с таблицей ArtGallery, с которой работаем на протяжении всей лабораторной работы.

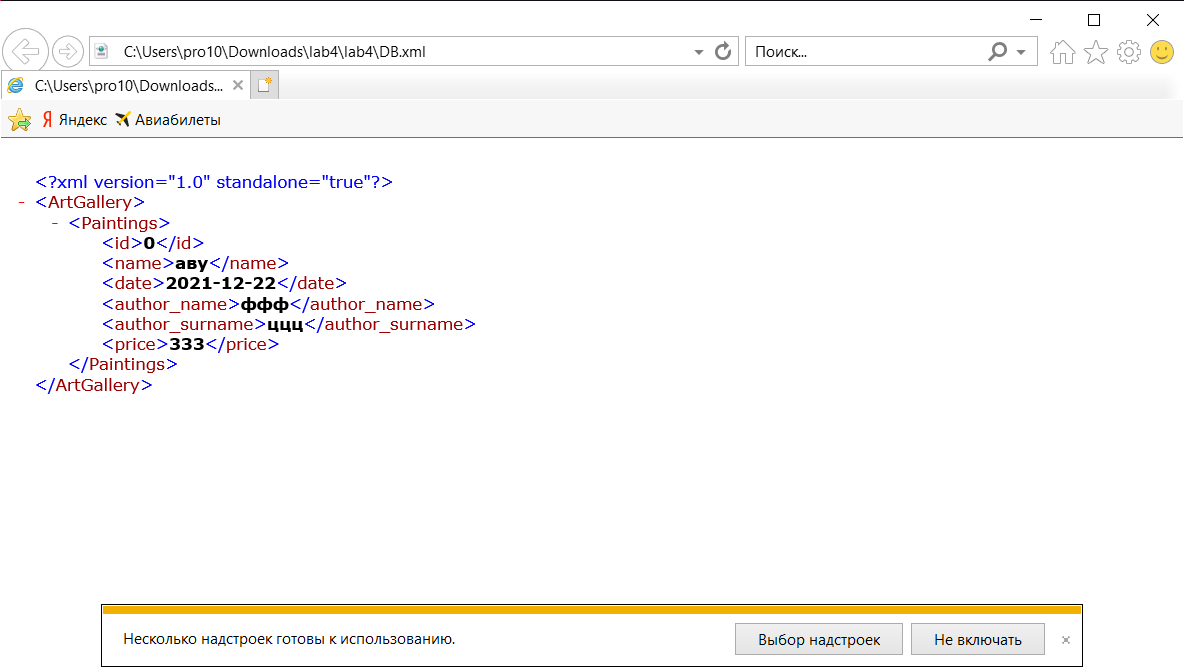
После этого была создана веб-форма – главная страница. Веб-форма позволяет просматривать список всех картин, а также имеет кнопки для сохранения/загрузки БД в/из .xml файл, а также кнопку для очистки резидентной базы данных.



*Рис. – Главная страница при пустой БД*

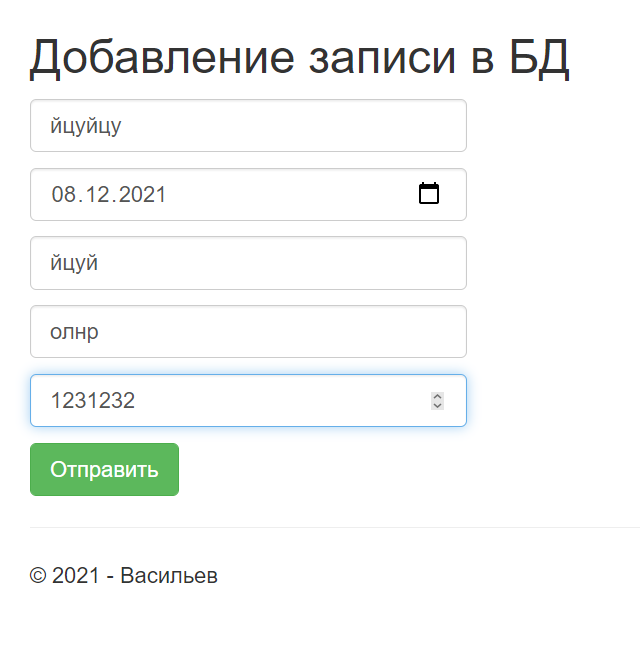


*Рис. – Главная страница после загрузки БД из XML*

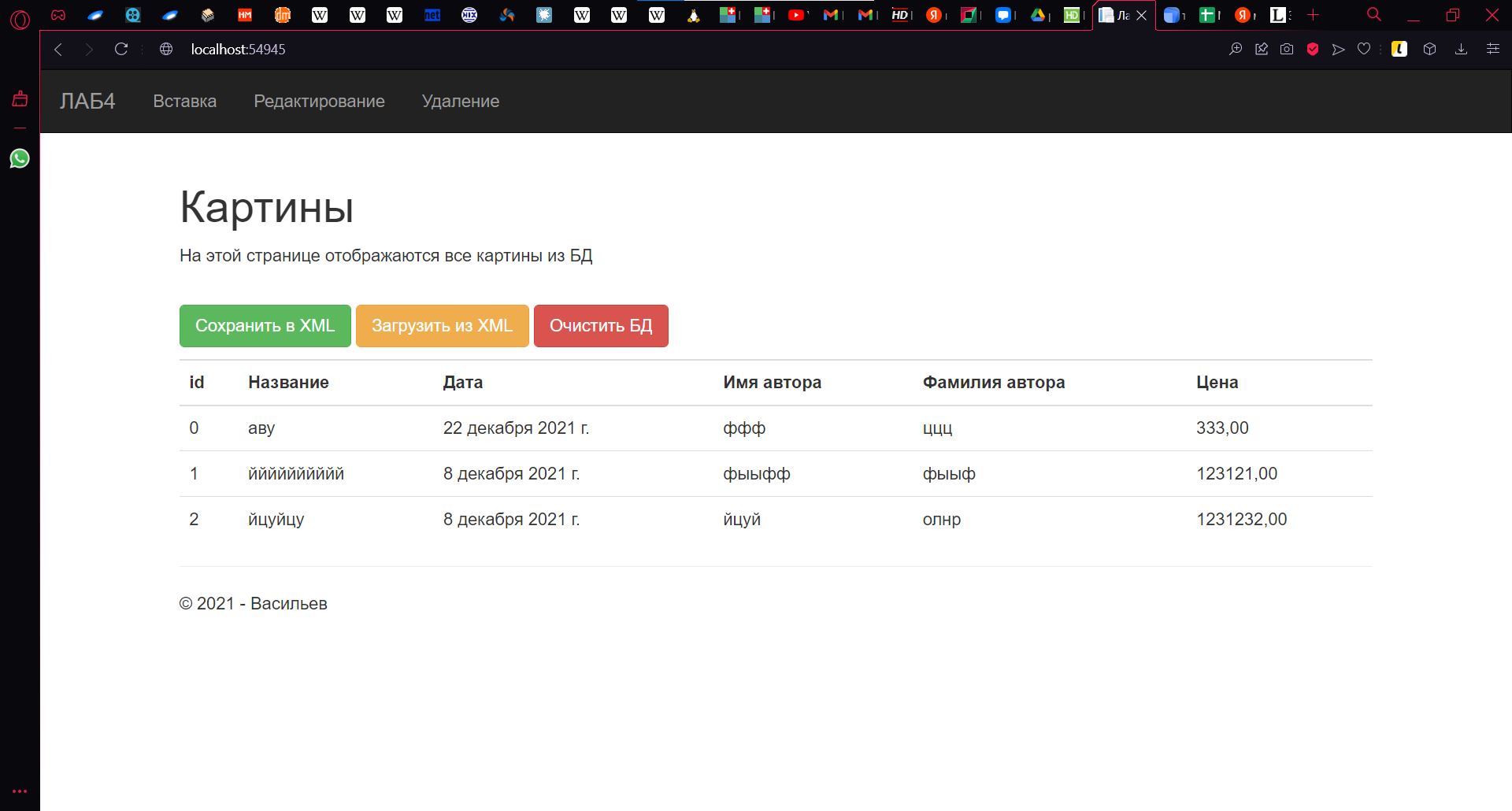
**

*Рис. – Содержимое файла XML*

Была создана веб-форма Insert для вставки новых строк в резидентную БД. На ней размещены 5 полей, необходимых для заполнения информации о картинах. Также размещена кнопка, по клику на которую, происходит обработка данных полей и вставка в таблицу. После успешной вставки происходит перенаправление на главную страницу.

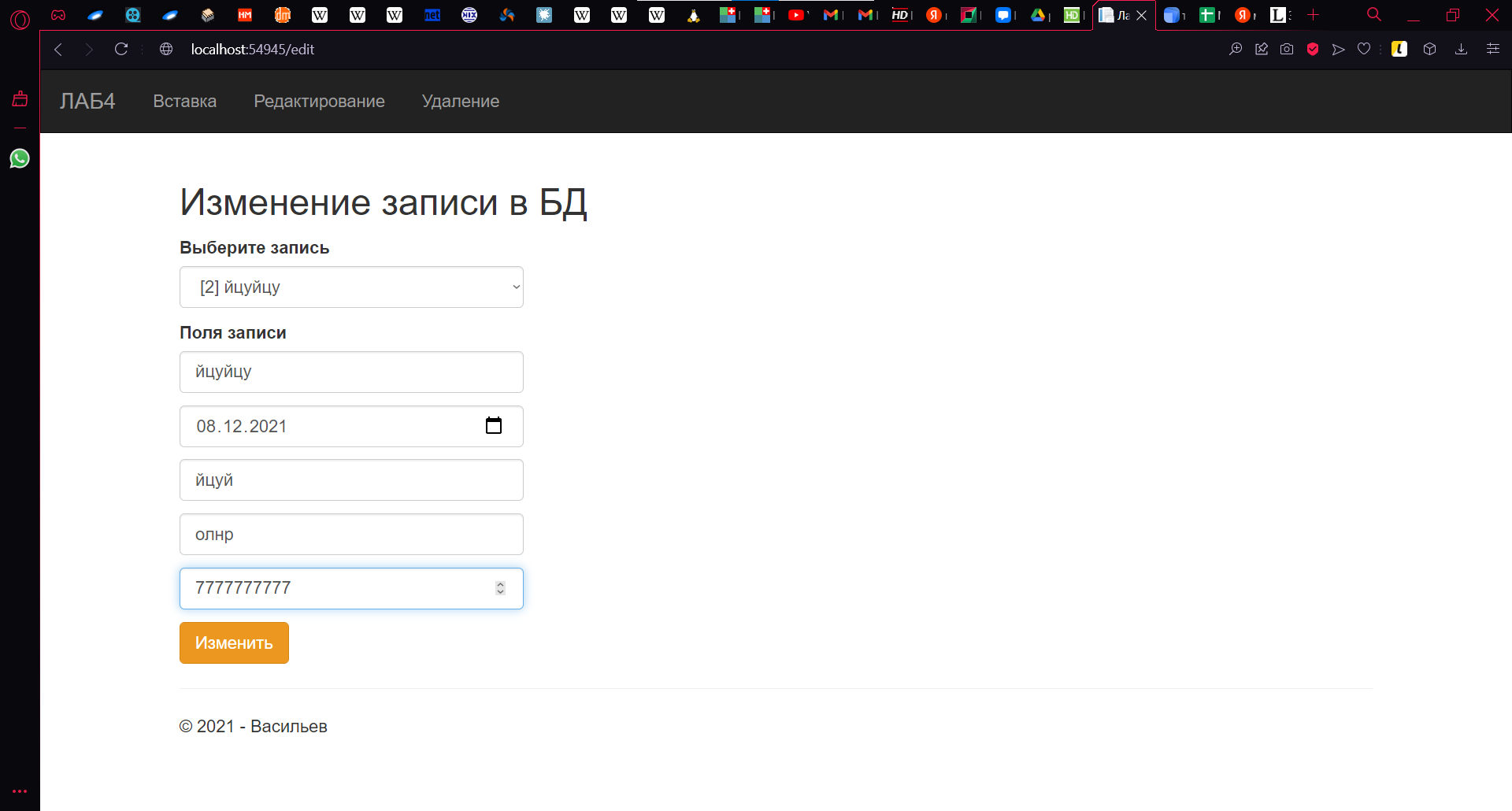


*Рис. – Страница вставки записи*

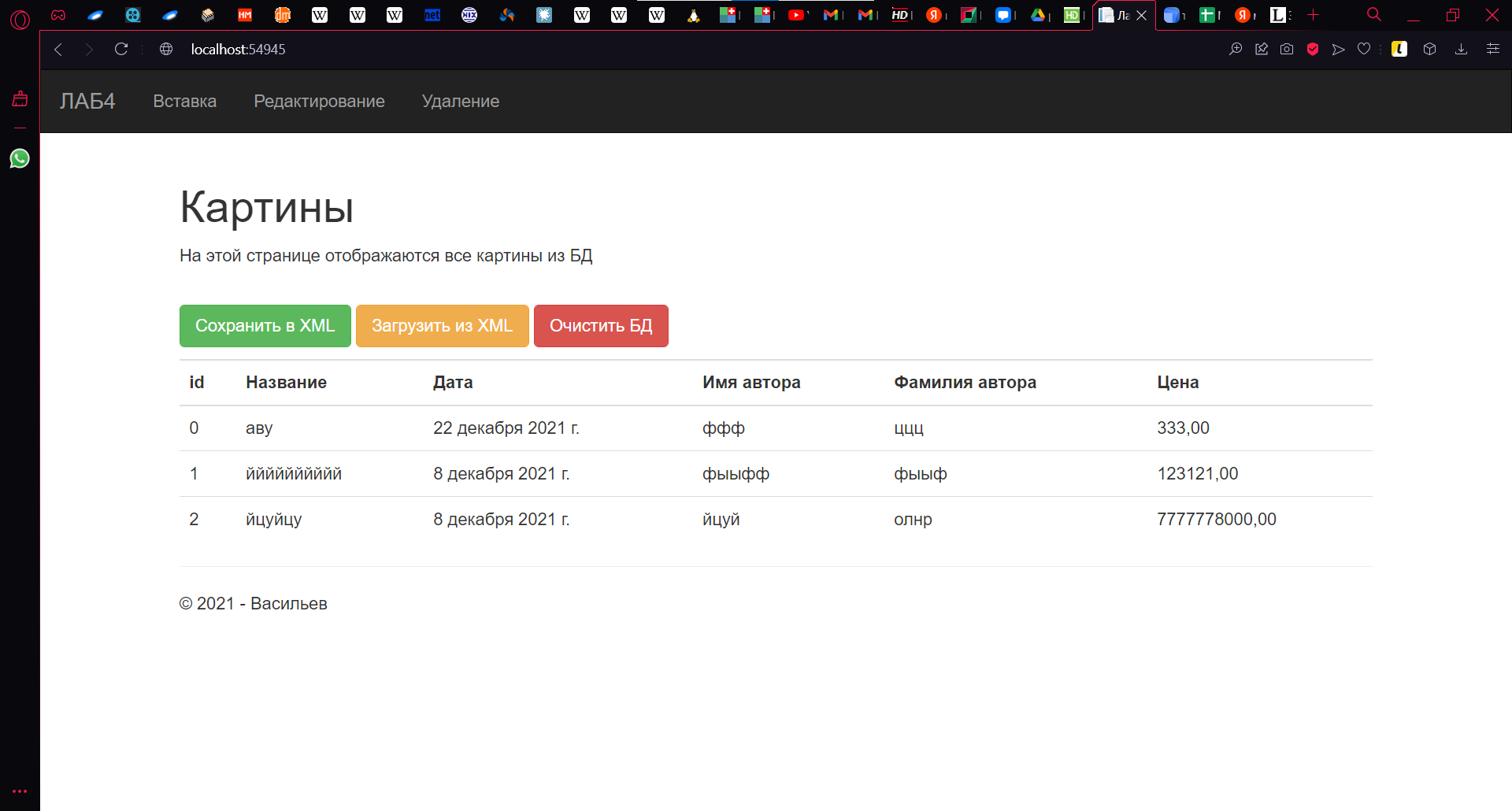
****

*Рис. – Таблица после вставки*

Была создана веб-форма Edit для редактирования существующих строк в БД. На ней размещен выпадающий список, при выборе значения которого меняются следующие 5 полей, необходимых для редактирования информации о картинах. Также размещена кнопка, по клику на которую, происходит обработка данных полей и изменение их в таблице.

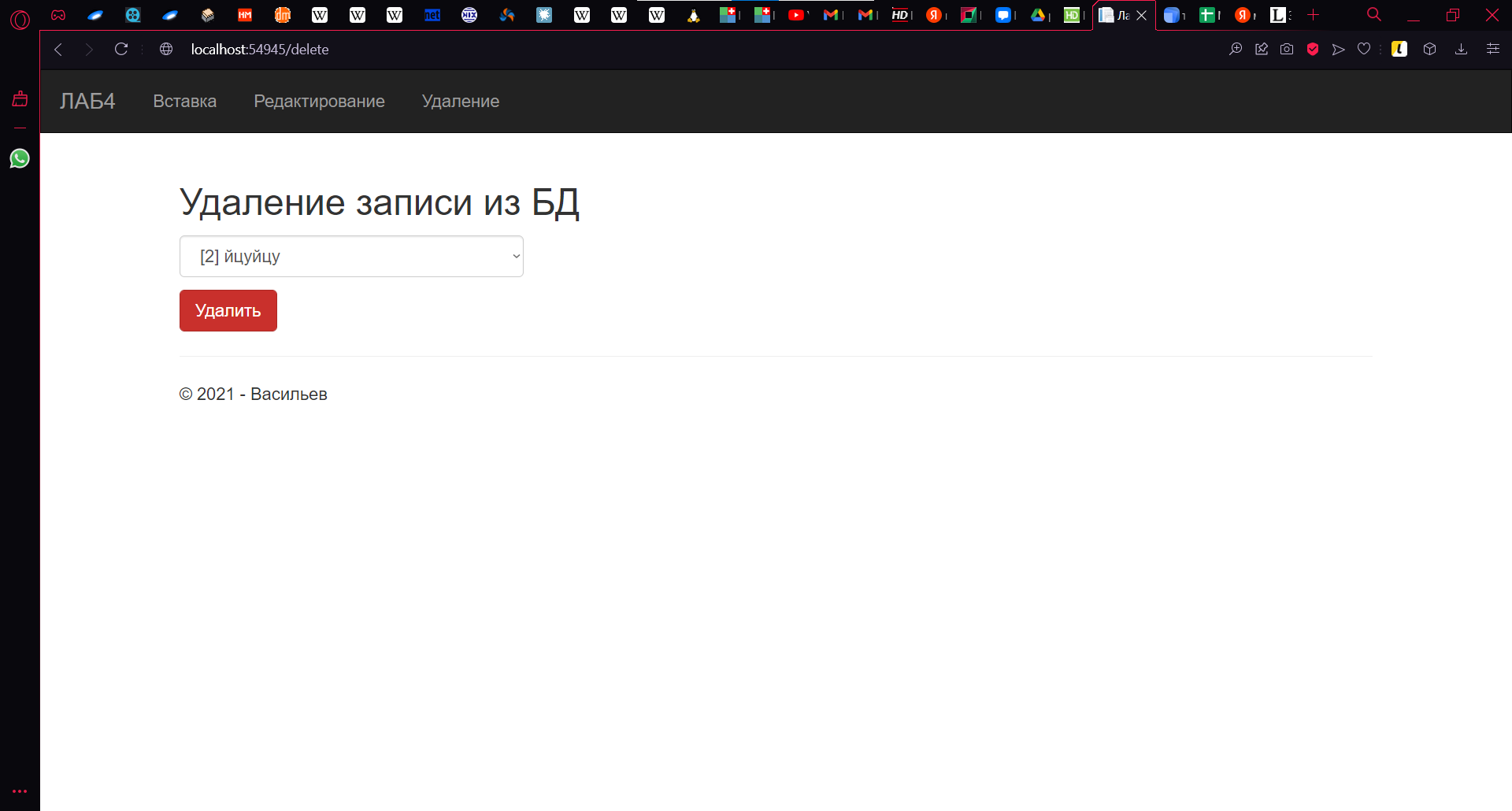


*Рис. – Страница редактирования записи*



*Рис. – Таблица после редактирования*

Была создана веб-форма Delete для удаления строк из БД. На ней размещен выпадающий список, который позволяет выбрать необходимую для удаления запись. Также размещена кнопка, по клику на которую, происходит удаление выбранной записи.



*Рис. – Страница удаления записи*

Ответы на вопросы

1. **Какие преимущества имеет XML перед другими языками описания данных?**

* Независимость от платформ. Язык XML позволяет обмениваться данными системам, базирующимся на разных платформах. XML-документ может быть создан и разобран как текстовый файл с помощью устаревших или встроенных языков программирования, в состав которых не входят специальные библиотеки для работы с XML.
* Поддержка производителями. Библиотеки для работы с XML созданы для всех ведущих языков программирования и популярных СУБД. Использование этих библиотек позволяет существенно уменьшить объем кодирования при разработке “шлюзов” между приложениями.
* Самодокументированность. XML-документ “читабелен” для человека. Кроме того, наличие внутри него описания данных позволяет создавать автоматические программы их обработки, например универсальные модули загрузки данных, поступающих из разных систем в единое хранилище.
* Иерархичность. Это ключевое свойство языка. В отличие от формата CSV (текстового файла с разделителем “;”), XML позволяет легко описывать сложные структуры данных с неограниченной вложенностью объектов.
* Объектность. Структура данных XML отлично сочетается с объектно-ориентированной моделью программирования. Каждый тег XML-документа может быть поставлен в соответствие классу или свойству класса обрабатывающей программы. С другой стороны, есть возможность описать в XML-формате каждый прикладной объект предметной области как отдельный тег.
* Расширяемость. В процессе эксплуатации XML-формата в него можно добавлять новые теги. Это не приведет к фатальному изменению структуры данных, просто читающие и пишущие программы нужно будет дополнить классами или функциями, распознающими эти теги.

1. **В чем отличия реляционных и XML моделей хранения данных?**

|  |  |
| --- | --- |
| **Реляционная модель** | **XML модель** |
| Табличное представление. | Иерархическое представление. |
| Строгая структура. К каждой строке таблицы применяется одна и та же схема. | Статические определения схемы. Не строго структурированная структура. Гибкое определение схемы. XML-схема может существовать для всех или некоторых XML-документов. Схемы легко расширяемы. |
| Все отношения определены первичными ключами и внешними ключами. | Документ содержит и данные, и информацию о связях. |
| Последовательность не имеет значения. Информация организована во множества, которые неупорядочены по определению. | Последовательность имеет значение. Информация организована в последовательности, которые упорядочены по определению |
| Жестко типизирована. Каждая колонка имеет строго один тип данных | Опционально типизирована. Типы могут быть определены для некоторых или для всех элементов и атрибутов в XML-схеме. |
| Стандартизация ANSI/ISO. | Стандартизация W3C. |
| 3-значная логика: true, false, null. | 2-значная логика: true, false. |
| NULL | Пустые элементы, отсутствующие элементы |

1. **В каких случаях следует отдавать предпочтение формату XML  при обмене данными между приложениями?**

Может быть лучшим хранение данных в XML, комбинируя их с реляционными данными, если для данных истинно одно или несколько следующих утверждений:

* Данные естественным образом отображаются в иерархический формат. Это противоположно данным, которые отображаются в табличном виде и удобно хранятся в реляционной базе данных. Иерархические данных может быть трудно отобразить на реляционную схему
* Схема часто трансформируется. Изменение бизнес-процессов, внедрение новых услуг или товаров или правительственные руководящие указания часто требуют обработки новых или других элементов. Поскольку XML схемы гибкие, можно посчитать практичным хранение XML-документы в их естественном формате вместе с существующими реляционными данными, чтобы избежать сложностей, которые могут возникнуть из-за частых изменений реляционной схемы.
* Преобразование схемы в общем случае легче осуществляется в XML, чем в реляционном формате. Некоторые реляционные изменения схемы просты, например, добавление колонки. Однако, некоторые довольно сложны, например, нормализация таблицы в несколько таблиц. В сложных случаях, вы можете сэкономить много времени и усилий, храня изменчивую часть данных в колонке типа XML.
* Данные имеют существенное количество атрибутов, редко имеющих значения. Такие атрибуты преобразуются в пустые ячейки в реляционной таблице. Поиск данных или другая аналитика в реляционных таблицах, которые содержат пустые ячейки, могут давать недостоверные или ошибочные результаты. Хранение данных в формате XML может помочь предотвратить такие ошибки. Некоторые приложения постоянно производят такие атрибуты, значения которых пусты или не определены. Данные часто содержат такие атрибуты, когда существует большое количество возможных атрибутов.
* Компоненты объекта имеют смысл в контексте только данного объекта. То есть, компоненты принадлежат объекту. Опасность заключается в нормализации данных до такой степени, что вам приходится соединять многочисленные столбцы при выполнении каждого запроса.

Выводы

В ходе выполнения лабораторной работы была освоена методика создания приложений ASP.NET с хранением данных в формате XML.

Приложение А. Листинг программы

Global.asax

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Optimization;

using System.Web.Routing;

using System.Web.Security;

using System.Web.SessionState;

namespace lab4

{

public class Global : HttpApplication

{

void Application\_Start(object sender, EventArgs e)

{

// Код, выполняемый при запуске приложения

RouteConfig.RegisterRoutes(RouteTable.Routes);

BundleConfig.RegisterBundles(BundleTable.Bundles);

PermanentCache.create\_dataset();

}

}

public class PermanentCache

{

static public DataSet artGallery = new DataSet("ArtGallery");

static public int id\_iter = 0;

static public string xmlPath = "C:\\Users\\Mike\\source\\repos\\lab4\\lab4\\DB.xml";

public static void create\_dataset()

{

DataTable paintingsTable = artGallery.Tables.Add("Paintings");

DataColumn pkPaintingID = paintingsTable.Columns.Add("id", typeof(Int32));

paintingsTable.Columns.Add("name", typeof(string));

paintingsTable.Columns.Add("date", typeof(string));

paintingsTable.Columns.Add("author\_name", typeof(string));

paintingsTable.Columns.Add("author\_surname", typeof(string));

paintingsTable.Columns.Add("price", typeof(double));

paintingsTable.PrimaryKey = new DataColumn[] { pkPaintingID };

}

}

}

Default.aspx

<%@ Page Language="C#" MasterPageFile="/Site.Master" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="lab4.Default" %>

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

<h1>Картины</h1>

<p>На этой странице отображаются все картины из БД</p>

<br />

<form id="form1" runat="server">

<p>

<asp:Button ID="SaveToXml" runat="server" Text="Сохранить в XML" CssClass="btn btn-success" OnClick="SaveToXml\_Click" />

<asp:Button ID="LoadFromXml" runat="server" Text="Загрузить из XML" CssClass="btn btn-warning" OnClick="LoadFromXml\_Click" />

<asp:Button ID="ClearDataSet" runat="server" Text="Очистить БД" CssClass="btn btn-danger" OnClick="ClearDataSet\_Click" />

</p>

<asp:Table ID="table" runat="server" CssClass="table" HorizontalAlign="Center"></asp:Table>

</form>

</asp:Content>

Default.aspx.cs

using System;

using System.Data;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace lab4

{

public partial class Default : System.Web.UI.Page

{

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

{

Print\_Table();

}

protected void Print\_Table()

{

table.Rows.Clear();

table.Rows.Clear();

DataTableReader reader = PermanentCache.artGallery.CreateDataReader();

int numcols = reader.FieldCount;

string[] headerPhrases = new string[] {

"id",

"Название",

"Дата",

"Имя автора",

"Фамилия автора",

"Цена"

};

TableHeaderRow header = new TableHeaderRow();

for (int k = 0; k < headerPhrases.Length; k++)

{

TableHeaderCell cell = new TableHeaderCell();

cell.Text = headerPhrases[k];

header.Cells.Add(cell);

}

table.Rows.Add(header);

while (reader.Read())

{

TableRow row = new TableRow();

for (int i = 0; i < numcols; i++)

{

TableCell cell = new TableCell();

string textContent;

if (headerPhrases[i] == "Дата")

{

DateTime date = DateTime.Parse(reader[i].ToString());

textContent = date.ToString("D");

}

else

if (headerPhrases[i] == "Цена")

{

float price = float.Parse(reader[i].ToString());

textContent = price.ToString("F");

}

else

{

textContent = reader[i].ToString();

}

cell.Text = textContent;

row.Cells.Add(cell);

}

table.Rows.Add(row);

}

reader.Close();

}

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

{

PermanentCache.artGallery.WriteXml(PermanentCache.xmlPath);

Page.Response.Redirect(Page.Request.Url.ToString(), true);

}

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

{

PermanentCache.artGallery.Clear();

PermanentCache.artGallery.ReadXml(PermanentCache.xmlPath);

PermanentCache.id\_iter = Convert.ToInt32(PermanentCache.artGallery.Tables["Paintings"].Compute("max([id])", string.Empty)) + 1;

Page.Response.Redirect(Page.Request.Url.ToString(), true);

}

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

{

PermanentCache.artGallery.Clear();

Page.Response.Redirect(Page.Request.Url.ToString(), true);

}

}

}

Insert.aspx

<%@ Page Language="C#" MasterPageFile="/Site.Master" AutoEventWireup="true" CodeBehind="insert.aspx.cs" Inherits="lab4.Insert" %>

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

<h1>Картины</h1>

<p>На этой странице отображаются все картины из БД</p>

<br/>

<form id="form1" runat="server">

<asp:Table ID="table" runat="server" CssClass="table" HorizontalAlign="Center"></asp:Table>

<br/><br/>

<h2>Добавление записи в БД</h2>

<p><asp:TextBox ID="name" runat="server" type="text" CssClass="form-control" placeholder="название" required="true"></asp:TextBox></p>

<p><asp:TextBox ID="date" runat="server" type="date" CssClass="form-control" placeholder="дата" required="true"></asp:TextBox></p>

<p><asp:TextBox ID="author\_name" runat="server" type="text" CssClass="form-control" placeholder="Имя автора" required="true"></asp:TextBox></p>

<p><asp:TextBox ID="author\_surname" runat="server" type="text" CssClass="form-control" placeholder="Фамилия автора" required="true"></asp:TextBox></p>

<p><asp:TextBox ID="price" runat="server" type="number" CssClass="form-control" placeholder="цена" required="true"></asp:TextBox></p>

<p><asp:Button ID="Button1" runat="server" Text="Отправить" CssClass="btn btn-success" OnClick="Button1\_Click"/></p>

</form>

</asp:Content>

Insert.aspx.cs

using System;

using System.Collections.Generic;

using System.Data;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace lab4

{

public partial class Insert : System.Web.UI.Page

{

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

{

Print\_Table();

}

protected void Print\_Table()

{

table.Rows.Clear();

table.Rows.Clear();

DataTableReader reader = PermanentCache.artGallery.CreateDataReader();

int numcols = reader.FieldCount;

string[] headerPhrases = new string[] {

"id",

"Название",

"Дата",

"Имя автора",

"Фамилия автора",

"Цена"

};

TableHeaderRow header = new TableHeaderRow();

for (int k = 0; k < headerPhrases.Length; k++)

{

TableHeaderCell cell = new TableHeaderCell();

cell.Text = headerPhrases[k];

header.Cells.Add(cell);

}

table.Rows.Add(header);

while (reader.Read())

{

TableRow row = new TableRow();

for (int i = 0; i < numcols; i++)

{

TableCell cell = new TableCell();

string textContent;

if (headerPhrases[i] == "Дата")

{

DateTime date = DateTime.Parse(reader[i].ToString());

textContent = date.ToString("D");

}

else

if (headerPhrases[i] == "Цена")

{

float price = float.Parse(reader[i].ToString());

textContent = price.ToString("F");

}

else

{

textContent = reader[i].ToString();

}

cell.Text = textContent;

row.Cells.Add(cell);

}

table.Rows.Add(row);

}

reader.Close();

}

protected void Button1\_Click(object sender, EventArgs e)

{

PermanentCache.artGallery.Tables["Paintings"].Rows.Add(new object[] { PermanentCache.id\_iter++, name.Text, date.Text, author\_name.Text, author\_surname.Text, price.Text });

Page.Response.Redirect(Page.Request.Url.ToString(), true);

}

}

}

Delete.aspx

<%@ Page Language="C#" MasterPageFile="/Site.Master" AutoEventWireup="true" CodeBehind="delete.aspx.cs" Inherits="lab4.Delete" %>

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

<form id="form1" runat="server">

<h2>Удаление записи из БД</h2>

<p><asp:DropDownList ID="selectedId" runat="server" AutoPostBack="true" CssClass="form-control"></asp:DropDownList></p>

<p><asp:Button ID="Button1" runat="server" Text="Удалить" CssClass="btn btn-danger" OnClick="Button1\_Click"/></p>

</form>

</asp:Content>

Delete.aspx.cs

using System;

using System.Collections.Generic;

using System.Data;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace lab4

{

public partial class Delete : System.Web.UI.Page

{

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

{

string selectedValue = selectedId.SelectedValue;

selectedId.Items.Clear();

DataTableReader reader = PermanentCache.artGallery.CreateDataReader();

int numcols = reader.FieldCount;

string itemDescription;

while (reader.Read())

{

itemDescription = String.Format("[{0}] {1}", reader[0].ToString(), reader[1].ToString());

ListItem lst = new ListItem(itemDescription, reader[0].ToString());

selectedId.Items.Add(lst);

}

reader.Close();

if (selectedValue != "") { selectedId.SelectedValue = selectedValue; }

}

protected void Button1\_Click(object sender, EventArgs e)

{

DataRow forDelete = PermanentCache.artGallery.Tables["Paintings"].Rows.Find(int.Parse(selectedId.SelectedValue));

PermanentCache.artGallery.Tables["Paintings"].Rows.Remove(forDelete);

Page.Response.Redirect(Page.Request.Url.ToString(), true);

}

}

}

Edit.aspx

<%@ Page Language="C#" MasterPageFile="/Site.Master" AutoEventWireup="true" CodeBehind="edit.aspx.cs" Inherits="lab4.Edit" %>

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

<form id="form1" runat="server">

<h2>Изменение записи в БД</h2>

<label>Выберите запись</label>

<p><asp:DropDownList ID="selectedId" runat="server" OnSelectedIndexChanged="selectedId\_SelectedIndexChanged" AutoPostBack="true" CssClass="form-control"></asp:DropDownList></p>

<label>Поля записи</label>

<p><asp:TextBox ID="name" runat="server" type="text" CssClass="form-control" placeholder="название" required="true"></asp:TextBox></p>

<p><asp:TextBox ID="date" runat="server" type="date" CssClass="form-control" placeholder="дата" required="true"></asp:TextBox></p>

<p><asp:TextBox ID="author\_name" runat="server" type="text" CssClass="form-control" placeholder="Имя автора" required="true"></asp:TextBox></p>

<p><asp:TextBox ID="author\_surname" runat="server" type="text" CssClass="form-control" placeholder="Фамилия автора" required="true"></asp:TextBox></p>

<p><asp:TextBox ID="price" runat="server" type="number" CssClass="form-control" placeholder="цена" required="true"></asp:TextBox></p>

<p><asp:Button ID="Button1" runat="server" Text="Изменить" CssClass="btn btn-warning" OnClick="Button1\_Click"/></p>

</form>

</asp:Content>

Edit.aspx.cs

using System;

using System.Data;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace lab4

{

public partial class Edit : System.Web.UI.Page

{

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

{

string selectedValue = selectedId.SelectedValue;

selectedId.Items.Clear();

DataTableReader reader = PermanentCache.artGallery.CreateDataReader();

int numcols = reader.FieldCount;

string itemDescription;

string selected\_Id = "";

while (reader.Read())

{

if (selectedId.Items.Count == 0) { selected\_Id = reader[0].ToString(); }

itemDescription = String.Format("[{0}] {1}", reader[0].ToString(), reader[1].ToString());

ListItem lst = new ListItem(itemDescription, reader[0].ToString());

selectedId.Items.Add(lst);

}

reader.Close();

if (selectedValue != "") { selectedId.SelectedValue = selectedValue; }

if (selected\_Id != "" && selectedValue == "") { Show\_Edit\_Form(selected\_Id); }

}

protected void Show\_Edit\_Form(string value)

{

int id = int.Parse(value);

DataRow selectedRow = PermanentCache.artGallery.Tables["Paintings"].Rows.Find(id);

name.Text = selectedRow["name"].ToString();

date.Text = selectedRow["date"].ToString();

author\_name.Text = selectedRow["author\_name"].ToString();

author\_surname.Text = selectedRow["author\_surname"].ToString();

price.Text = selectedRow["price"].ToString();

}

protected void Button1\_Click(object sender, EventArgs e)

{

int id = int.Parse(selectedId.SelectedValue);

DataRow[] dr = PermanentCache.artGallery.Tables["Paintings"].Select("id=" + id);

if (dr != null)

{

dr[0]["name"] = name.Text;

dr[0]["date"] = date.Text;

dr[0]["author\_name"] = author\_name.Text;

dr[0]["author\_surname"] = author\_surname.Text;

dr[0]["price"] = price.Text;

}

Page.Response.Redirect(Page.Request.Url.ToString(), true);

}

protected void selectedId\_SelectedIndexChanged(object sender, EventArgs e)

{

Show\_Edit\_Form(selectedId.SelectedValue);

}

}

}