**МИНОБРНАУКИ РОССИИ**

**г. Новосибирск**

**Федеральное государственное автономное образовательное учреждение высшего образования «Новосибирский национальный исследовательский государственный университет, НГУ»**

**Структурное подразделение Новосибирского государственного университета - Высший колледж информатики Университета**

Кафедра Информатики\_\_\_\_

**КУРСОВОЙ ПРОЕКТ**

**(РАБОТА)**

по дисциплине «WEB-программирование»

на тему «Создание сайта-библиотеки с выставками картин»

**Выполнил:**

Студент Прилепко Дмитрий Владимирович

Специальность Программирование в компьютерных системах

Группа 703б2

**Проверил:**

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_**\_\_\_**

(подпись и

расшифровка подписи)

Оценка: «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2020 г.

**Содержание**

1. СОДЕРЖАНИЕ………………………………………………………………………………… 2
   1. Актуальность………………………………………………………………………. 3
   2. Цель работы……………………………………………………………………….. 5
   3. Задачи……………………………………………………………………………… 5
2. ТЕОРИТИЧЕСКАЯ ЧАСТЬ………………………………………………………………….... 6
3. ПРАКТИЧЕСКАЯ ЧАСТЬ…………………………………………………………………….. 7
4. ПРИЛОЖЕНИЕ 1. ……………………………………………………………………………...10
5. ПРИЛОЖЕНИЕ 2. …………………………………………………………………………….. 12

**Введение**

**Актуальность**

Актуальность темы курсовой работы обоснована тем, что порой не все люди имеют возможность посещать временные выставки и галереи музеев. Рабочий график, религиозные причины, семейные обязанности, быт и другие дела, которые занимают время и не дают человеку добавить в свой график дел и обязанностей посещение выставочных мероприятий музеев который в ту же очередь имеют свои собственные графики выставок, мероприятий и программ, так же бывает, что порой просто не предоставляется возможностью попасть в музеи, к примеру по причине географической отдаленности музея от места проживания человека, или же закрытие музея на ремонт, так же и не плановые реставрационные работы в связи которых может перенестись выставка экспозиций или же в крайнем случае вовсе её отмена. А также бывает то, что человек не в состоянии здоровья посетить выставочные мероприятия по причине наличия физических травм, инвалидности, психологических расстройств и других заболеваний.

Имеется и проблема с отсутствием пиара у не крупных музеев из-за чего происходит так, что некоторые малые музеи часто затмеваются тенью более крупных музеев в связи с чем у первых заметно ниже средняя посещаемость чем у вторых, это приводит к тому что некоторые люди вовсе не знают о тех малых музеях и об их предстоящих выставках поскольку они не упоминаются в большинстве своем в средствах массовой информации.

Основной возраст аудитории, посещающей музей, составляет от 25 до 54 лет с разными уровнями образования что говорит о малой заинтересованности в посещение со стороны современного молодого посетителя, так же основная причина походов в музеи носит случайный характер и происходит в основном в форме групповых и индивидуальных туристических экскурсий.

Все вышеперечисленное подводит к тому, что в нынешнее время часто не предоставляется возможным и порой не комфортным посещение музеев для того, чтобы человек мог и был заинтересован в живую лицезреть произведения искусства в постоянной экспозиции музеев или же временные выставки, проходящие в них.

**Цель работы** в разработки сайта-библиотеки с выставками картин состоит в следующем:

* Изучить frontend составляющею при разработке web-сайта
* Изучение предметной области для реализации проекта
* Составить список функций, который должен поддерживать сервис
* Изучение способа разработки интернет ресурса сайта-библиотеки с выставками картин
* Изучить backend составляющею при разработке web-сайта
* Научиться пользоваться фреймворками и библиотеками

Достижение поставленной цели потребовало решения следующих **задач**:

1. Реализации общей структуры сайта:
2. Добавление CSS и Bootstrap на страницу:
3. Создание Json файла с массивом объектов основного контента
4. Написание JavaScript файла:
5. Написание консольного приложения на C# для имитации работы сервера

**Теоретическая часть**

**План работы:**

* Составить общую структуру сайта
* Разработать дизайн сайта
* Описать общею схему работы сайта
* Составить ряд функций необходимых для вывода экспозиций на сайт
* Разработать серверную часть сайта

Пользователю должна предоставляться возможность просмотра экспозиций сайте, так же должна иметься возможность авторизации администрации сайта для модерирования показываемого контента.

**Практическая часть**

Общая устройство клиентской части представляет собой связь двух страниц, первая вступительная страница с приветствием которая переадресовывает пользователя на основную страницу имеющею структуру (см. рис. 1) состоящею из 8 основных блоков:

1. Блок c названием сайта.
2. Блок краткого описания сайта для пользователя
3. Блок логотипа сайта
4. Блок для вывода рекламной информации на сайте
5. Блок названия выбранного раздела в меню пользователем
6. Основной блок для вывода контента
7. Блок для авторизации администрации сайта
8. Блок с контактной информацией

Пользовательская часть была выполнена с использованием:

1. Html
2. CSS
3. JavaScript
4. Bootstrap
5. jQuery
6. AngularJS
7. Pug

Серверная часть была выполнена использованием:

1. C# с использованием пространства имен System.Net
2. Ajax
3. AngularJS

**Общее устройство серверной составляющей:**

Когда пользователь при вводе логина и пароля производит попытку авторизации (см. рис. 2) путем нажатия кнопки Login, происходит запуск функции Login, которая создает запрос с отправкой введенных данных логина и пароля на сервер, после чего начинается работа функции Request\_response на сервере которая принимает запрос, производит обработку полученных сервером данных о логине и пароля пользователя, после чего производит отправку ответа на страничку на попытку авторизации, после которой имеется два варианта дальнейших сценариев:

1. Успешная попытка с результатом авторизации (см. рис. 3).
2. Не успешная попытка авторизации с выводом сообщения пользователю об этом (см. рис. 5)

Вариации использования сайта представлены на схеме (см. рис. 4).

**Основное устройство клиентской составляющий** состоит из вызова функции readJSON которая в ту очередь производит запрос для загрузки json файла с передачей пути к нему и получением его для дальнейшего обрабатывания его данных необходимых для выполнения следующий вызываемой функции Load которая в своем время использует обработанные данные из функции readJSON и производит с ними работу во время которой создаются секции контейнеры в которых происходит добавление блоков с полученным контентом из json файла, а так же производиться вывод рекламы на страницу с помощью функции Adver.

**Приложение 1.** **Рисунки, Схемы**

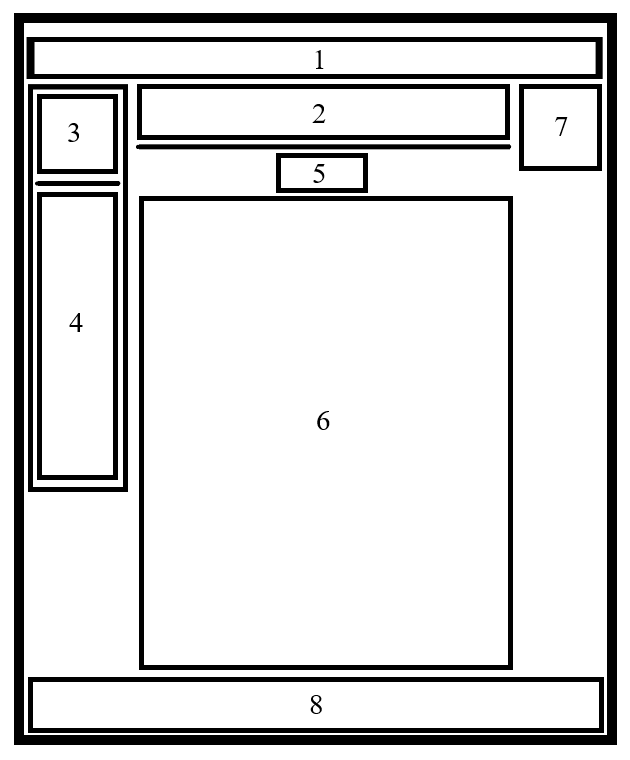


Рисунок - Общая структура сайта

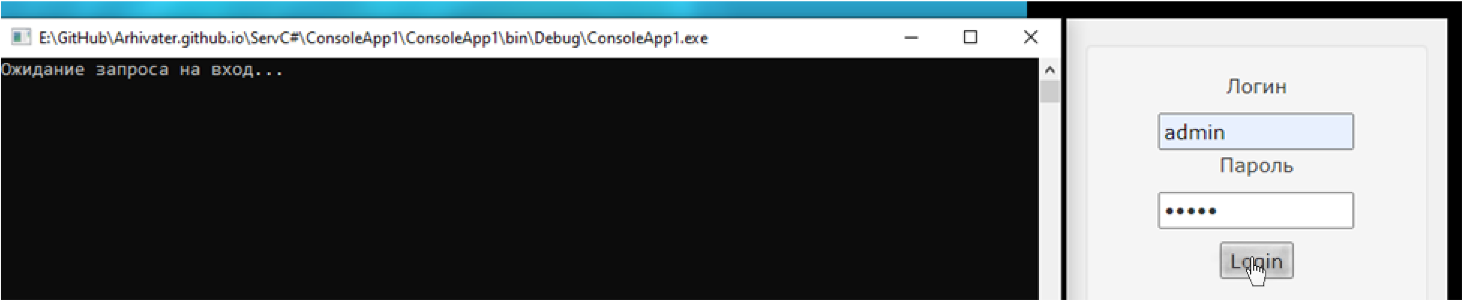


Рисунок 2 – Попытка авторизации

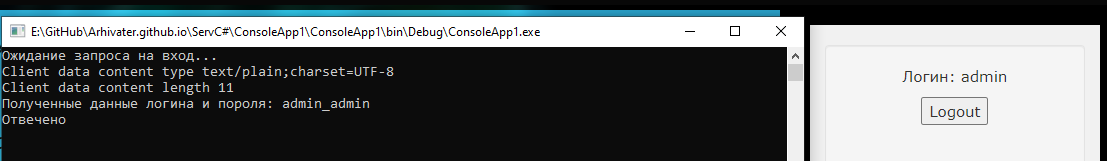


Рисунок 3 – Результат авторизации

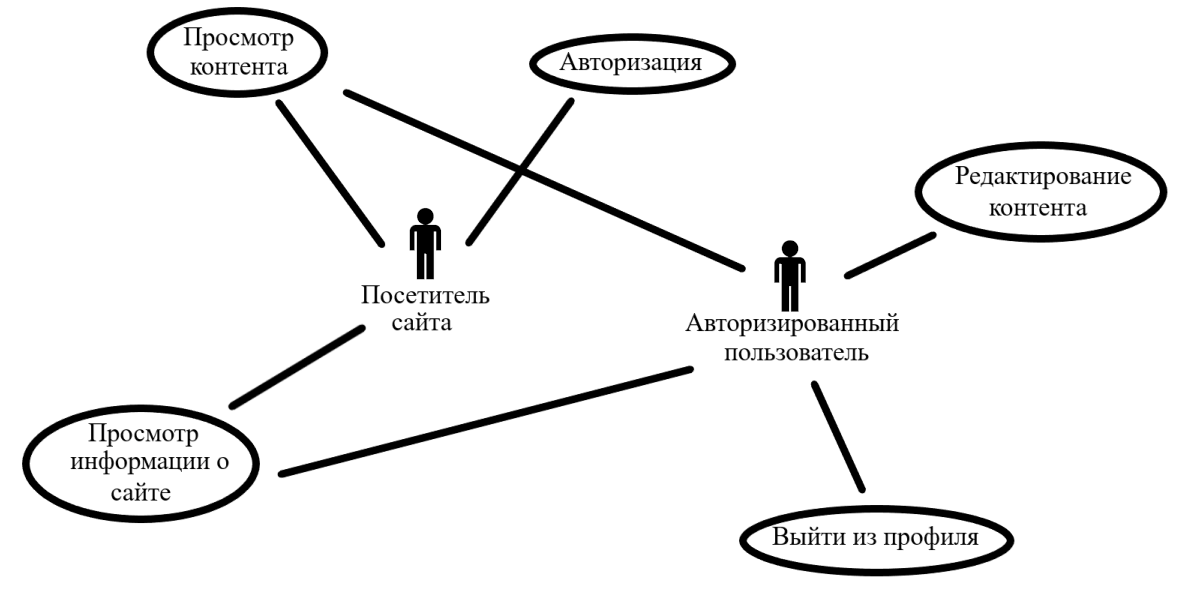


Рисунок 4 – Общая схема вариаций использования сайта

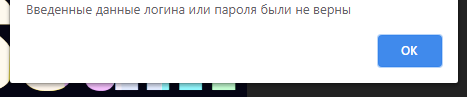


Рисунок 5 – Не успешная попытка авторизации

**Приложение 2. Листинг кода**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Museum</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js">

</script>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">

</script>

<script type="text/javascript" src="https://code.jquery.com/jquery-3.3.1.min.js"></script>

<link rel="stylesheet" href="Museum\_CSS.css">

<script src="Museum.js"></script>

</head>

<body class="style1">

<div class="navbar navbar-inverse style2">

<div class="container-fluid fig">

<img src="Logo2\_Museum.png">

</div>

</div>

<div class="container-fluid text-center">

<div class="row content">

<div class="col-sm-1 sidenav style3">

<p class="text-align: center;">

<img src="Logo\_Museum.png" class="style4">

</p>

<p>

<h3 class="header\_h3">Museum</h3>

</p>

<hr>

<div id="Advertising">

<script>Adver();</script>

</div>

</div>

<div class="col-sm-9 text-left">

<div class="Background\_Start">

<h1 class="Welcome">Добро пожаловать в Museum</h1>

<h3>Перед вами «Museum», здесь вы можете посмотреть на картины художников всего мира, которые содержаться в музеях и прочесть об авторах этих картин, а также вы можете узнать дату написания просматриваемых вами картин</h3>

</div>

<hr>

<div class="Title">

<p class="fig style5"><b class="Title\_text">Картины</b>

</div>

<div id="Main\_section" class="w3-row-padding">

</div>

</div>

<div class="col-sm-2 sidenav style6">

<div id="Login\_Museum" class="well style7">

<p id="lg" >Логин</p>

<input class="style8" type="text" id="POSTparam1" value="">

<p id="password">Пароль</p>

<input class="style8" type="password" id="POSTparam2" value="">

<p id="Login1" class="fig style9">

<input id="Login2" type="button" onclick="Login(this.value)" value="Login">

</p>

</div>

</div>

</div>

</div>

<footer class="container-fluid text-center style10">

<p>Контактная информация</p>

</footer>

</body>

</html>

**Код Museum.js**

var kappa="";

var JSON\_var="";

var Museum="";

var path="./List.json";

function readJSON(path) {

var xhr = new XMLHttpRequest();

xhr.open('GET', path, true);

xhr.responseType = 'blob';

xhr.onload = function(e) {

if (this.status == 200) {

kappa = new File([this.response], 'temp');

var fileReader = new FileReader();

fileReader.addEventListener('load', function(){

kappa=fileReader.result;

JSON\_var = JSON.stringify(kappa);

Museum = JSON.parse(kappa.substring(0, kappa.length - 1));

Load();

});

fileReader.readAsText(kappa);

}

}

xhr.send();

}

readJSON(path);

function Load() {

Clear\_Main\_section = document.getElementById('Main\_section');

Clear\_Main\_section.innerHTML = '';

var amount = Museum.Paintings.length;

var shelf = Math.ceil(amount/3);

var k = 0;

var Main\_section = document.getElementById('Main\_section');

for(let g = 1; g <= shelf ; g++) {

var section = document.createElement("div");

section.classList.add("w3-row-padding");

section.classList.add("w3-section");

section.id=("Paintings\_section"+'\_'+g);

Main\_section.appendChild(section);

for(var i = 0; i < 3; i++)

{

var ins = "";

ins +=

'<div class="w3-third w3-container w3-margin-bottom">'+

'<img class="img\_Paintings" src="'+Museum.Paintings[k].Paintings\_img+'" alt="Norway" style="width:100%" class="w3-hover-opacity">'+

'<div class="w3-container w3-white">' +

'<div class="media">' +

'<div class="media-left" style="padding-top: 35px;">'+

'<img class="Author" src="'+Museum.Paintings[k].Authors\_img+'" class="media-object">'+

'</div>'+

'<div class="media-body" style="padding-top: 5px;">'+

'<h4 class="media-heading">'+Museum.Paintings[k].Paintings\_name+'</h4>'+

'<h4><small><i>'+Museum.Paintings[k].Paintig\_date+'</i></small></h4>'+

'<p>'+Museum.Paintings[k].Author\_name+'</p>'+

'<p>'+Museum.Paintings[k].Paintig\_description+'</p>'

'</div>'+

'</div>'+

'</div>'+

'</div>';

if (k < amount){

k+=1;

}

var id\_section = "Paintings\_section"+'\_'+g

document.getElementById(id\_section).innerHTML += ins;

}

}

}

function Login(str) {

var name = "";

if (str.length == 0) {

return;

} else {

var xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

var key = this.responseText;

if (key == "true"){

alert("Добро пожаловать " + name + " !");

document.getElementById("POSTparam1").remove();

document.getElementById("POSTparam2").remove();

document.getElementById("password").remove();

document.getElementById("Login1").remove();

document.getElementById("lg").remove();

var base = document.getElementById("Login\_Museum");

var Log = document.createElement("p");

Log.id=("Logo\_name");

base.appendChild(Log);

document.getElementById("Logo\_name").innerHTML="Логин: " + name;

var Dp = document.createElement("p");

Dp.id = ("")

base.appendChild(Dp);

var Logout = document.createElement("input");

Logout.setAttribute('type', 'button');

Logout.id=("Logout\_Museum");

Logout.setAttribute('alt', 'Painting\_Image');

Logout.setAttribute('value', 'Logout');

Logout.setAttribute('onclick', 'Logout()');

base.appendChild(Logout);

} else {

document.getElementById("POSTparam1").value="";

document.getElementById("POSTparam2").value = "";

alert("Введенные данные логина или пароля были не верны");

}

}

};

xmlhttp.open("POST", "http://localhost:1111", true);

var send ="";

name = document.getElementById("POSTparam1").value;

send = document.getElementById("POSTparam1").value;

send += "\_"+document.getElementById("POSTparam2").value

xmlhttp.send(send);

}

}

function Logout() {

document.getElementById("Logout\_Museum").remove();

document.getElementById("Logo\_name").remove();

var base = document.getElementById("Login\_Museum");

var plg = document.createElement("p");

plg.id=("lg");

base.appendChild(plg);

document.getElementById("lg").innerHTML="Логин";

var inp1 = document.createElement("input");

inp1.setAttribute('type', 'text');

inp1.setAttribute('value', '');

inp1.id=("POSTparam1");

inp1.setAttribute('class', 'style8');

base.appendChild(inp1);

var pps = document.createElement("p");

pps.id=("password");

base.appendChild(pps);

document.getElementById("password").innerHTML="Пароль";

var inp2 = document.createElement("input");

inp2.setAttribute('type', 'password');

inp2.setAttribute('value', '');

inp2.id=("POSTparam2");

inp2.setAttribute('class', 'style8');

base.appendChild(inp2);

var plg2 = document.createElement("p");

plg2.id=("Login1");

plg2.setAttribute('class', 'fig');

plg2.setAttribute('class', 'style9');

base.appendChild(plg2);

var base2 = document.getElementById("Login1");

var inp3 = document.createElement("input");

inp3.setAttribute('type', 'button');

inp3.setAttribute('value', '');

inp3.id=("Login2");

inp3.setAttribute('value', 'Login');

inp3.setAttribute('onclick', 'Login(this.value)');

base2.appendChild(inp3);

}

function Adver () {

for (var l = 0; l<4; l++) {

$('#Advertising').after(function() {

var vst = "<a href='https://georgevorob.github.io/index.html'>"+

"<a href='https://georgevorob.github.io/index.html'>"+

"<p class='text-align: center;'>"+

"<img src='Advertising.png' alt='Здесь могла быть ваша реклама...'>"+

"</p>"+

"</a>"+

"<br>";

return vst;

});

}

}

**Код Museum\_CSS.css**

.navbar {

margin-bottom: 0;

border-radius: 0;

}

.row.content {

height: 921px

}

.sidenav {

padding-top: 20px;

background-color: #f1f1f1;

height: 100%;

}

.Welcome {

text-align: center;

}

footer {

background-color: #555;

color: white;

padding: 15px;

}

@media screen and (max-width: 767px) {

.sidenav {

height: auto;

padding: 15px;

}

.row.content {height:auto;}

}

.img\_Paintings{

width: 439px;

height: 375px;

}

.header\_h3 {

color: #46b823;

text-align: center;

font-size: 30px;

}

.Museum\_h3 {

color: #46b823;

text-align: center;

font-size: 40px;

}

.glyphicon glyphicon-log-in {

height: 10px;

}

.buttons\_header{

text-align: center;

}

.fig {

text-align: center;

}

.Author {

width: 45px;

height: 70px;

}

.Background\_Start {

color: #000000;

font: italic 30px "Italic",monospace;

border: 5px solid Black;

padding: 5px;

opacity: 0.8;

background: #33ccff;

text-align: left;

}

.Title {

font: italic 30px "Italic",monospace;

border: 5px solid Black;

padding: 5px;

margin-left: 40%;

margin-right: 40%;

opacity: 0.8;

background: #33ccff;

text-align: left;

}

.Title\_text {

text-shadow: 1px 1px 2px black, 0 0 1em red;

color: black;

font-size: 2em;

}

hr {

border: none;

background-color: #33ccff;

color: #33ccff;

height: 5px;

}

.btn-success {

width: 100px;height: 34px;

margin-left: 10px;

margin-right: 10px;

}

.style1{

background: url(Background\_1.png) no-repeat;

background-position: center center;

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

background-color: #F4A460;

border-collapse: collapse;

object-fit: cover;

}

.style2{

border-top-width: 15px;

border-bottom-width: 15px;

}

.style3{

height: 880px;

}

.style4{

height: 130px;

width: 130px;

}

.style5{

color: #0CFF00;

}

.style6{

height: 280px;

top: 20px;

right: 20px;

left: 10px;

width: 290px;

}

.style7{

height: 240px;

}

.style8{

width: 149px;

}

.style9{

padding-top: 10px;

}

.style10{

text-align: right;

margin-top: 29px;

}

**Код List.json**

{

"Paintings": [

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/8/83/Leonardo\_da\_Vinci\_-\_Portrait\_of\_a\_Musician.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/b/ba/Leonardo\_self.jpg/150px-Leonardo\_self.jpg",

"Paintings\_name": "«Портрет музыканта»",

"Paintig\_date": "1490-2",

"Author\_name": "Леона́рдо ди сер Пье́ро да Ви́нчи",

"Paintig\_description": "Авторство Леонардо да Винчи оспаривается редко, хотя и существует предположение, что автором портрета является не Леонардо, а Антонелло да Мессина."

},

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/e/ec/Mona\_Lisa%2C\_by\_Leonardo\_da\_Vinci%2C\_from\_C2RMF\_retouched.jpg/800px-Mona\_Lisa%2C\_by\_Leonardo\_da\_Vinci%2C\_from\_C2RMF\_retouched.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/b/ba/Leonardo\_self.jpg/150px-Leonardo\_self.jpg",

"Paintings\_name": "«Портре́т госпожи́ Ли́зы дель Джоко́ндо»",

"Paintig\_date": "варьируется между 1503 и 1505 годами",

"Author\_name": "Леона́рдо ди сер Пье́ро да Ви́нчи",

"Paintig\_description": "Одно из самых известных произведений живописи. Считается, что на картине изображена Лиза Герардини, супруга флорентийского торговца шёлком Франческо дель Джокондо."

},

{

"Paintings\_img": "https://sun9-58.userapi.com/c845416/v845416689/826f4/8R0ZQXbkNXQ.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/1/1f/Thomas\_Cole.jpg",

"Paintings\_name": "«Крушение»",

"Paintig\_date": "1836 год",

"Author\_name": "Томас Коул",

"Paintig\_description": "Одно из картин входящих в серию из 5 картин Томаса Коула под названием <<Путь Империи>>."

},

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/e/ea/Van\_Gogh\_-\_Starry\_Night\_-\_Google\_Art\_Project.jpg/757px-Van\_Gogh\_-\_Starry\_Night\_-\_Google\_Art\_Project.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/7/74/VincentVanGoghFoto.jpg/150px-VincentVanGoghFoto.jpg",

"Paintings\_name": "«Звёздная Ночь»",

"Paintig\_date": "1889 год",

"Author\_name": "Винсент ван Гог",

"Paintig\_description": "Представляет вид из восточного окна спальни Ван Гога в Сен-Реми-де-Прованс на предрассветное небо и вымышленную деревню."

},

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/f/f4/The\_Scream.jpg/470px-The\_Scream.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/f/f0/Edvard\_Munch\_1912.jpg/200px-Edvard\_Munch\_1912.jpg",

"Paintings\_name": "«Крик»",

"Paintig\_date": "в промежутке между 1893 и 1910 годами",

"Author\_name": "Эдвард Мунк",

"Paintig\_description": "Cерия картин норвежского художника-экспрессиониста Эдварда Мунка. На них изображена кричащая в отчаянии человеческая фигура на фоне кроваво-красного неба и крайне обобщённого пейзажного фона."

},

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/7/73/God2-Sistine\_Chapel.png/800px-God2-Sistine\_Chapel.png",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/5/5e/Miguel\_%C3%81ngel%2C\_por\_Daniele\_da\_Volterra\_%28detalle%29.jpg/518px-Miguel\_%C3%81ngel%2C\_por\_Daniele\_da\_Volterra\_%28detalle%29.jpg",

"Paintings\_name": "«Сотворение Адама»",

"Paintig\_date": "около 1511 года",

"Author\_name": "Микела́нджело Буонарро́ти",

"Paintig\_description": "Фреска Микеланджело, является четвёртой из девяти центральных композиций потолка Сикстинской капеллы, посвящённых девяти сюжетам книги Бытия. Фреска иллюстрирует эпизод: <<И сотворил Бог человека по образу Своему>>"

},

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/1/1e/Shishkin%2C\_Ivan\_-\_Morning\_in\_a\_Pine\_Forest.jpg/800px-Shishkin%2C\_Ivan\_-\_Morning\_in\_a\_Pine\_Forest.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/Ivan\_Shishkin\_restoration.jpg/250px-Ivan\_Shishkin\_restoration.jpg",

"Paintings\_name": "«Утро в сосновом лесу»",

"Paintig\_date": "1889 год",

"Author\_name": "И. И. Шишкин, К. А. Савицкий",

"Paintig\_description": "Савицкий написал медведей. но коллекционер П. М. Третьяков стёр его подпись, так что автором картины часто указывается один Шишкин.Картина пользуется популярностью благодаря композиционному включению в пейзажное полотно элементов анималистической сюжетности. Работа детально передает состояние природы, увиденное художником на острове Городомля. Показан не глухой дремучий лес, а солнечный свет, пробивающийся сквозь колонны высоких деревьев. Чувствуют приближение утра резвящиеся медвежата."

},

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/4/4a/Hovhannes\_Aivazovsky\_-\_The\_Ninth\_Wave\_-\_Google\_Art\_Project.jpg/800px-Hovhannes\_Aivazovsky\_-\_The\_Ninth\_Wave\_-\_Google\_Art\_Project.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/c/c6/Aivazovsky\_-\_Self-portrait\_1874.jpg/546px-Aivazovsky\_-\_Self-portrait\_1874.jpg",

"Paintings\_name": "«Девятый вал»",

"Paintig\_date": "1850 год",

"Author\_name": "Айвазовский Иван Константинович",

"Paintig\_description": "Живописец изображает море после очень сильного ночного шторма и людей, потерпевших кораблекрушение. Лучи солнца освещают громадные волны. Самая большая из них — девятый вал — готова обрушиться на людей, пытающихся спастись на обломках мачты."

},

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/e/ec/Karl\_Brullov\_-\_The\_Last\_Day\_of\_Pompeii\_-\_Google\_Art\_Project.jpg/800px-Karl\_Brullov\_-\_The\_Last\_Day\_of\_Pompeii\_-\_Google\_Art\_Project.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/4/4f/Brjullov.jpg/512px-Brjullov.jpg",

"Paintings\_name": "«После́дний день Помпе́и»",

"Paintig\_date": "1833 год",

"Author\_name": "Брюллов, Карл Павлович",

"Paintig\_description": "На картине изображены события в Помпеях во время катастрофического извержения Везувия, которое произошло в 79 году нашей эры."

},

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/Apotheosis.jpg/800px-Apotheosis.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/8/8f/Vasili\_Vereshchagin.jpg/432px-Vasili\_Vereshchagin.jpg",

"Paintings\_name": "«Апофеоз войны»",

"Paintig\_date": "1871 год",

"Author\_name": "Василий Верещагин",

"Paintig\_description": "На раме сделана надпись: «Посвящается всем великим завоевателям — прошедшим, настоящим и будущим». Изначально полотно называлось «Торжество Тамерлана». Замысел был связан с Тамерланом, войска которого оставляли за собой такие пирамиды черепов, однако картина не носит конкретно-исторический характер."

},

{

"Paintings\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/e/e1/John\_William\_Waterhouse\_-\_Fair\_Rosamund.jpg/444px-John\_William\_Waterhouse\_-\_Fair\_Rosamund.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/b/ba/John\_William\_Waterhouse\_001.jpg/376px-John\_William\_Waterhouse\_001.jpg",

"Paintings\_name": "«Прекрасная Розамунда»",

"Paintig\_date": "около 1150 года",

"Author\_name": "Джон Уильям Уотерхаус",

"Paintig\_description": "На картине изображена дочь Уолтера Фиц-Ричарда де Клиффорд и Маргариты де Тосни, любовница короля Англии Генриха II Плантагенета. За свою красоту получила прозвища «Прекрасная Розамунда»"

},

{

"Paintings\_img": "http://ayvazovskiy.su/wp-content/uploads/2017/02/ledgor.jpg",

"Authors\_img": "https://upload.wikimedia.org/wikipedia/commons/thumb/c/c6/Aivazovsky\_-\_Self-portrait\_1874.jpg/546px-Aivazovsky\_-\_Self-portrait\_1874.jpg",

"Paintings\_name": "«Ледяные горы»",

"Paintig\_date": "1870 год",

"Author\_name": "Айвазовский Иван Константинович",

"Paintig\_description": "Полотно Айвазовского Ледяные горы является известным на весь мир, автор написал, чтобы увековечить великое открытие отечественных мореходов, нашедших континент на самом юге Земли."

}

]

};

**Код консольного приложения (.NET Framework) ConsoleApp1**

using System;

using System.Net;

using System.IO;

using System.Threading.Tasks;

namespace server

{

class Program

{

private static async Task Request\_response()

{

HttpListener listener = new HttpListener();

listener.Prefixes.Add("http://localhost:1111/");

listener.Start();

Console.WriteLine("Ожидание запроса на вход...");

while (true)

{

string send = "";

string responseString = "";

HttpListenerContext context = await listener.GetContextAsync();

HttpListenerRequest request = context.Request;

HttpListenerResponse response = context.Response;

response.Headers.Add("Access-Control-Allow-Origin", "\*");

if (request.HttpMethod == "POST")

{

if (!request.HasEntityBody)

{

Console.WriteLine("No client data was sent with the request.");

}

System.IO.Stream body = request.InputStream;

System.Text.Encoding encoding = request.ContentEncoding;

System.IO.StreamReader reader = new System.IO.StreamReader(body, encoding);

if (request.ContentType != null)

{

Console.WriteLine("Client data content type {0}", request.ContentType);

}

Console.WriteLine("Client data content length {0}", request.ContentLength64);

send = reader.ReadToEnd();

Console.WriteLine("Полученные данные логина и пороля: " + send);

if (send == "admin\_admin")

{

responseString = "true";

}

else { responseString = "false"; }

}

Console.WriteLine("Отвечено");

byte[] buffer = System.Text.Encoding.UTF8.GetBytes(responseString);

response.ContentLength64 = buffer.Length;

Stream output = response.OutputStream;

output.Write(buffer, 0, buffer.Length);

output.Close();

}

}

static async Task Main(string[] args)

{

await Request\_response();

}

}

}