Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра экономической информатики

КОНТРОЛЬНАЯ РАБОТА

по дисциплине

«Системный анализ и проектирование информационных ресурсов»

Выполнил: Проверила:

студент гр. 472302 Куц Е.Н. ст. преподаватель Атрощенко Н. А.

Минск 2016

**ИНСТРУМЕНТАРИИ, ИСПОЛЬЗОВАННЫЕ В РАБОТЕ**

1.Общие требования

1.1 Использование блочной верстки с использованием 2-3 видов позиционирования.





#leftPartMain{

width: 50%;

height: 100%;

margin: 0 left;

position: absolute;

text-align: center;

display:inline-block;

vertical-align: middle;

padding-top: 10%;

padding-bottom: 15%;

}

#rightPart{

width: 50%;

height:100%;

margin: 0 auto;

position: fixed;

left:50%;

text-align: center;

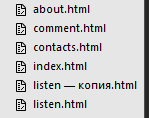
display:inline-block;

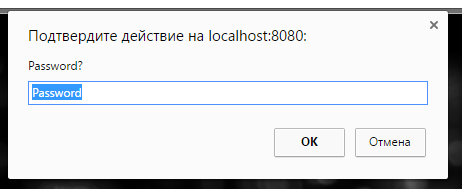
padding-top: 25vh;

padding-bottom: 25vh;

}

1.2. Количество веб-страниц – 5.

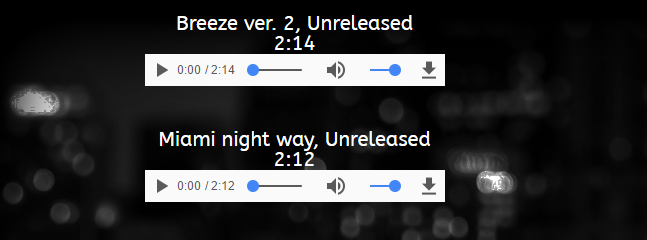


1.3 Авторизация с помощью JavaScript.  


2. CSS3, HTML5

2.1 Использование CSS3 и HTML5, в том числе:

API audio;



for(var i=0;i<data.songs.length;i++){

console.log(data.songs.length);

                $('.lmenu').append('<li>' + data.songs[i].title + ', <b>'+data.songs[i].album+'</b><br>'+data.songs[i].duration+'<br> <audio controls><source src="../resource/sound/'+data.songs[i].link +'" type="audio/mpeg"></audio><br><br><br></li>');

                }

API форм;



<form>

<input type="text" id="name" placeholder=" Name" class="but bg-1 button--size-l button--text-upper button--text-thin button--border-medium button--winona anyCase"></input><br><br><span>focus fire</span></p>

<textarea id="comment" placeholder="Comment" class="but bg-1 button--size-l button--text-upper button--text-thin button--border-medium button--winona anyCase"></textarea><br><br>

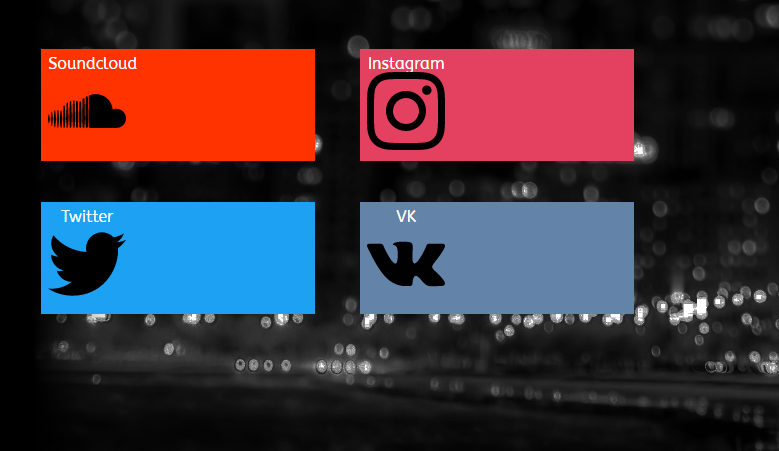
<input type="button" id="addrec" value=" Leave comment " onclick=" addRecord();" class="but bg-3 button--size-l button--text-upper button--text-thin button--border-medium button--winona"></input><br><br>

<input type="reset" value=" Clear form " class="but bg-3 button--size-l button--text-upper button--text-thin button--border-medium button--winona"></input><br><br>

<input type="button" id="showcomm" value=" Show comment " onclick="showRecords();" class="but bg-3 button--size-l button--text-upper button--text-thin button--border-medium button--winona"></input><br><br><br><br>

</form>

API SVG;



img src="../resource/logo/soundcloud.svg" alt="soundcloud">

2.2 Использование CSS reset.

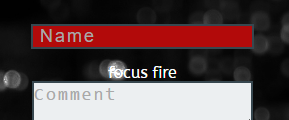
Файл «reset.css».

4. Применение динамического HTML.

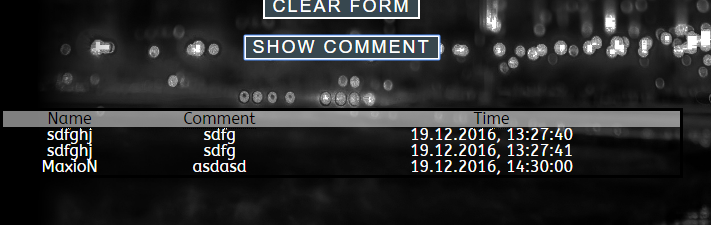
4.1. Перехват события наведения мыши.



4.2 Перехват события фокуса(+ Фильтры атрибутов, + Событие focus)



4.3 Вставка дочерних элементов (Jquery).



function showRecords(){

$("#showTable").html('');

$("#showTable").css('border', '3px solid black');

$("#showTable").append( $('<tr><td><a class="tooltip" href="#">Name<span class="classic">Name of commentator</span></a></td><td><a class="tooltip" href="#">Comment<span class="classic">Directly comment</span></a></td><td><a class="tooltip" href="#">Time<span class="classic">Commenting time</span></a></td></tr>'));

db.transaction(function(tx) {

tx.executeSql("SELECT \* FROM Comments", [], function(tx, result) {

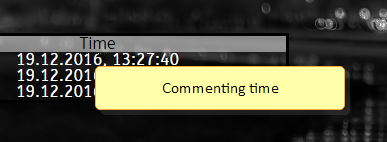
for(var i = 0; i < result.rows.length; i++) {

$("#showTable").append( $('<tr><td>'+ result.rows.item(i)['name'] +'</td><td>'+result.rows.item(i)['comment']+'</td><td>'+result.rows.item(i)['date']+'</td></tr>'));

}}, null)});

5. Применение библиотек.

5.1. Использование динамических элементов (всплывающая подсказка)



/\*ВСПЛЫВАЮЩИЕ ПОДСКАЗКИ\*/

.tooltip {

border-bottom: 1px dotted #000000;

color: #000000; outline: none;

cursor: help; text-decoration: none;

position: relative;

}

.tooltip span {

margin-left: -999em;

position: absolute;

text-decoration: none;

}

.tooltip:hover span {

font-family: Calibri, Tahoma, Geneva, sans-serif;

position: absolute;

left: 1em;

top: 2em;

z-index: 99;

margin-left: 0;

width: 250px;

}

.tooltip:hover em {

font-family: Candara, Tahoma, Geneva, sans-serif;

font-size: 1.2em;

font-weight: bold;

display: block;

padding: 0.2em 0 0.6em 0;

}

.classic { padding: 0.8em 1em; }

.custom { padding: 0.5em 0.8em 0.8em 2em; }

<strong>\* html</strong> a:hover { background: transparent; }

.classic { background: #FFFFAA; border: 1px solid #FFAD33;

border-radius: 5px 5px 0 0;

-moz-border-radius: 5px;

-webkit-border-radius: 5px;

box-shadow: 5px 5px 5px rgba(255, 255, 255, 0.1);

-webkit-box-shadow: 5px 5px rgba(255, 255, 255, 0.1);

-moz-box-shadow: 5px 5px rgba(255, 255, 255, 0.1);}

.critical { background: #FFCCAA; border: 1px solid #FF3334; }

5.2 Использвоание интерактивной анимации.

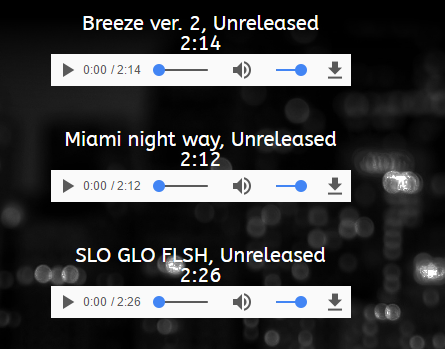


.logo{margin-top: 10%;margin-bottom: 10%; transition: 0.5s;}

.logo:hover{filter: drop-shadow(0 0 1vh #4F4F4F);}

6. Ajax+Jquery

6.1 Загрузка данных их *json*-файла.



function showContent(){

//Считываем через getJSON

$.getJSON('json/songs.json', function(data) {

//Считываем и записываем в таблицу

/\*$('#users').append('<tr><td>Дата основания</td><td>Сайт</td><tr>');

$('#users').append('<tr><td>' + data.found + "</td><td><a href='http://www." + data.site + "'> "+ data.site +" </a></td>"); \*/

//Считываем и записываем в список

for(var i=0;i<data.songs.length;i++){

console.log(data.songs.length);

                $('.lmenu').append('<li>' + data.songs[i].title + ', <b>'+data.songs[i].album+'</b><br>'+data.songs[i].duration+'<br> <audio controls><source src="../resource/sound/'+data.songs[i].link +'" type="audio/mpeg"></audio><br><br><br></li>');

                }

});

var cont = document.getElementById('contentBody');

var loading = document.getElementById('loading');

//Считываем и записываем в id="par"

$("#contentBody").html($("#loading").html());

$("#par").html(name);

var http = createRequestObject(); // создаем ajax-объект

if( http ) {

http.open('GET', link); // инициируем загрузку страницы

http.onreadystatechange = function () { // назначаем асинхронный обработчик события

if(http.readyState == 4) {

cont.innerHTML = http.responseText; // присваиваем содержимое

}

}

http.send(null);

} else {

document.location = link; // если ajax-объект не удается создать, просто перенаправляем на адрес

}

}

Songs.json:

{

"songs": [{

"title": "Breeze ver. 2",

"album": "Unreleased",

"duration": "2:14",

"link": "MaxioN - Breeze ver. 2.mp3"

},

{

"title": "Miami night way",

"album": "Unreleased",

"duration": "2:12",

"link":"MaxioN - Miami night way.mp3"

},

{

"title": "SLO GLO FLSH",

"album": "Unreleased",

"duration": "2:26",

"link":"MaxioN - SLO GLO FLSH.mp3"

},

{

"title": "Under your feet",

"album": "Unreleased",

"duration": "2:04",

"link":"MaxioN - Under your feet.mp3"

},

{

"title": "Number 1",

"album": "Unreleased",

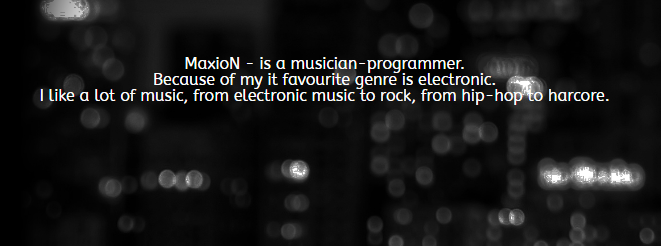
"duration": "1:09",

"link":"MaxioN - Number 1.mp3"

}]

}

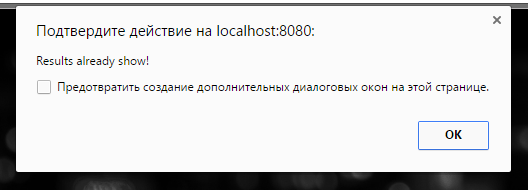
6.2. Подгрузка данных из html-файла.



showAbout();

$("#text").load('loadpages/aboutText.html');

6.3 Использование события click();



$( "#showcomm" ).bind( "click", function() {

alert( "Results already show!");

});

7. Модульная концепция AMD.

7.1. Использование ReueryJS, файл config.js:

requirejs.config({

paths: {

"jquery" : "jquery",

"script" : "script",

"showabout" : "showabout",

"auth" : "auth"

}

});

7.2 Модуль auth.js:

define(function(){

return function(){

var password=prompt('Password?','Password')

if(password=='watchback'){

alert('Right password!');

sessionStorage.setItem('pass',1);

}

else(alert('Wrong password!'));

location.reload();

}

});

7.3Использование модуля auth.js:

function auth(){

if(sessionStorage.getItem('pass')!=1){

require(['auth'], function(auth){

auth();

});}

}

auth();

8. Сервер. Используется node.js. Также используется Express.

8.1 Конфигурационный файл для запуска сервера server.js собственными настройками:

// grab express

var express = require('express');

// create an express app

var app = express();

// create an express route for the home page

// http://localhost:8080/

app.get('/', function(req, res) {

res.sendFile(\_\_dirname + '/index.html');

});

app.use(express.static(\_\_dirname));

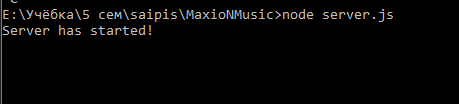
// start the server on port 8080

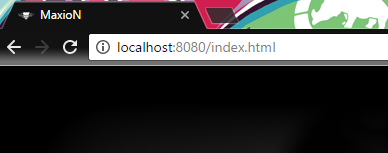
app.listen(8080);

// send a message

console.log('Server has started!');

8.2 Запуск приложения через сервер:





**Дополнительно** используется WebSQL для хранения и подгрузки данных.