Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Инженерно-экономический факультет

Кафедра экономической информатики

Дисциплина: Системный анализ и проектирование информационных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к контрольной работе

на тему

**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ**

Студент: гр. 572303 Шевяков М. А.

Руководитель: Атрощенко Н.А.

Минск 2017

**ЛИСТИНГ КОДА И СКРИНШОТЫ ОСНОВНЫХ ТРЕБОВАНИЙ К ПРИЛОЖЕНИЮ**

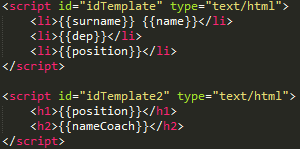
****

Рисунок 1 – Применение шаблонов для организации вывода данных

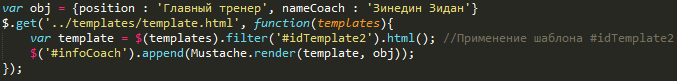


Рисунок 2 – Использование шаблонизатора Mustache

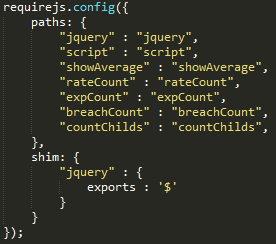


Рисунок 3 – Файл «config.js» - определение путей к модулям



Рисунок 4 – Реализация модульной концепции AMD (использование RequireJS)

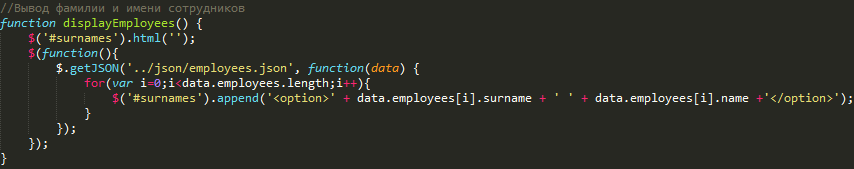


Рисунок 5 – Подгрузка данных в формате JSON



Рисунок 6 – Использование Ajax+jQuery. Привязка событий

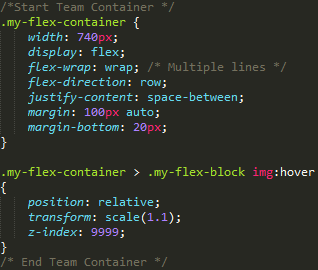


Рисунок 7 – Использование гибкой блочной вёрстки (flexbox)

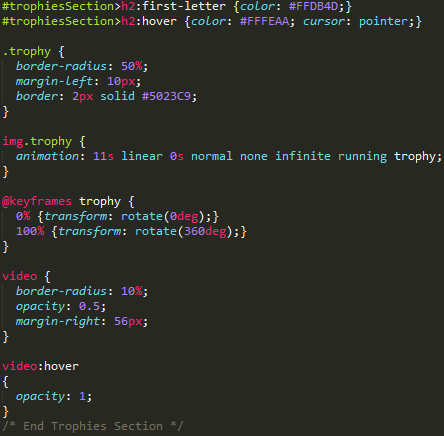


Рисунок 8 – Использование анимации, видео. Изменение видимости элементов

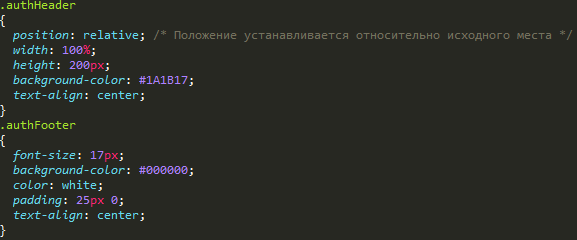


Рисунок 9 – Применение позиционирования элементов

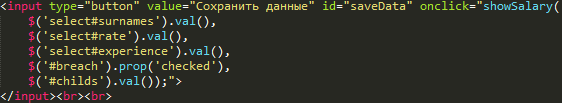


Рисунок 10 – Использование атрибутов val, prop

**Файл «db.js» -** создание MongoClient для подключения к базе данных MongoDB.

var MongoClient = require('mongodb').MongoClient;

var state = {

db : null

};

exports.connect = function (url, done){

if (state.db) {

return done();

}

MongoClient.connect(url, function(err, db) {

if (err) {

return done(err);

}

state.db = db;

done();

})

}

exports.get = function() {

return state.db;

}

**Файл «server.js» -** Создание HTTP-сервера на Node.js, использование npm, framework express. Подключение к MongoDB, добавление данных в базу.

var express = require('express');

var bodyParser = require('body-parser');

var db = require('./db');

var path = require('path');

var alert = require('alert-node');

var hbs = require("hbs");

var userMenu = 0;

var obj;

var app = express(); // Создали сервер

app.engine('html', require('ejs').renderFile);

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

app.use(bodyParser.json()); // Правильный парсинг JSON, который передали в body

app.use(bodyParser.urlencoded({ extended: true })); // Правильный парсинг данных формы

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

generateMenu();

res.render('index.html', obj);

})

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

res.render('index.html', obj);

})

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

res.sendFile(path.join(\_\_dirname) + '/html/authentication.html');

})

app.post('/authentication', function(req,res) {

//db.get().dropDatabase();

var userLogin = req.body.login;

var userPassword = req.body.password;

var inputValue = req.body.actionButton;

if (userLogin == "" || userPassword == "") {

alert('Логин или пароль заданы неверно!');

return res.sendFile(path.join(\_\_dirname) + '/html/authentication.html');

} else {

if (inputValue == "Войти") {

db.get().collection('users').findOne({login : userLogin, password : userPassword}, function (err, user) {

if (err) return res.sendStatus(500);

if (!user) {

alert("Неверный логин или пароль!");

return res.sendFile(path.join(\_\_dirname) + '/html/authentication.html');

}

alert("Добро пожаловать " + userLogin);

userMenu = 1;

generateMenu();

res.render('index.html', obj);

})

} else if (inputValue == "Зарегистрироваться"){

db.get().collection('users').insertOne({login : userLogin, password : userPassword}, function(err, user) {

if (err) return res.sendStatus(500);

alert("Пользователь " + userLogin + " зарегистрирован");

})

}

}

})

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

res.render('trophies.html', obj);

})

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

res.render('team.html', obj);

})

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

res.render('uniform.html', obj);

})

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

res.render('employees.html', obj);

})

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

alert("Выход успешен!");

userMenu = 0;

generateMenu();

res.render('index.html', obj);

})

app.post('/add',function(req,res) {

db.get().collection('employees').insertOne({name : req.body.name, salary : req.body.salary}, function(err, user) {

if (err) return res.sendStatus(500);

})

})

db.connect('mongodb://localhost:27017/mydatabase', function(err) {

if (err) {

return console.log(err);

} else {

console.log('База данных успешно подключена!');

}

app.listen(3012, function() {

console.log('Сервер успешно запущен!');

})

})

function generateMenu() {

if (userMenu == 0)

obj = {menu1 : '', menu2 : '', menu3 : '<a class="authClass" href="authentication">Войти</a>'}

else

obj = {menu1 : '<a href="employees">Руководство</a>', menu2 : '<a href="exit">Выйти</a>', menu3 : ''}

}

**Вывод:** в результате проделанной контрольной работы было создано функционирующее клиент-серверное веб-приложение. Основные требования, предъявляемые к проекту, были соблюдены и выполнены. Используя современные технологии по вёрстке макета (Bootstrap, flexbox и т.д.), а также применяя современные языки программирования (HTML5, DHTML, CSS JavaScript) спроектирован шестистраничный сайт для футбольной команды Real Madrid. В качестве сервера использовался Node.js, хранилищем данных выступает MongoDB.