**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования «КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт Вычислительной математики и информационных

технологий

Кафедра анализа данных и исследования операций

Направление 38.03.05 «Бизнес-информатика»

КУРСОВАЯ РАБОТА

**Сайт спортивных событий с использованием API сторонних Интернет-ресурсов**

Студент 3 курса

Группы 09 -601

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20 \_\_\_г. Балаев А.Ш.

Научный руководитель

\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_

уч. степень уч. звание

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_г. Мухтарова Т.М.

Казань – 2019

**СОДЕРЖАНИЕ**

1. Введение
2. Цель курсовой работы
3. Описание программы
4. Архитектура приложения
5. Разработка программы
6. Заключение
7. Список Литературы

Введение:

Свою курсовую я решил реализовать с помощью такого языка программирования как PHP версии 7.

Язык программирования PHP я решил выбрать, потому что у него низкий порог вхождения и он очень удобен для работы с бэкэндом. PHP – является серверным языком. У него есть свои плюсы и минусы.

Так же в ходе курсовой работы я использовал уже всем известный и стандартный набор таблиц каскадных стилей CSS и язык разметки HTML.

Реализовывать курсовую работу я начал с установки OpenServer’а.

Open Server – это серверная платформа, которая была создана для веб-программирования. Очень удобная платформа, с помощью которой можно реализовать проекты любого масштаба. Начиная от маленьких задач, в промежутке курсовых работ и заканчивая большими проектами.

У Open Server есть много конкурентов, я работ с большим количеством аналогов. Такие как денвер, apahce. Даже встроенный сервер в такое среде разработке как PHPSHTORM. Довольно интересная среда разработки. В начале своего обучения веб-программирования я выбрал именно его. Он требовал большой производительности ноутбука.

Свой первый веб-проект – по предмету в КФУ я как раз таки реализовывал в среде разработке – PHPSHTORM. Но в ходе работы у меня возникли трудности, поэтому я решил перейти на Open Server.

Также Open Server предоставляет работу с такими базами данными как phpMyAdmin, MySql менеджер, SQlite менеджер.

Мой выбор остановился на phpMyAdmin. Очень удобная база данных, с которой я также работал до курсовой работы. Со своими первыми веб-проектами. Есть также хороший конкурент как SQL express server.

У phpMyAdmin очень удобный и простой интерфейс для понимая.

Моя курсовая работа связана с интернет темой.

На сегодняшний день интернет – это очень быстро развивающаяся и удобная площадка для любых сфер деятельности.

Сфера деятельности моей курсовой – это интернет магазин.

Интернет магазин – это сайт предоставляющий пользователям приобрести той или иной продукт интернет магазина. Продукт - подразумевает собой не только одежду, но и программное обеспечение.

В настоящее время всё стремится к тому, чтобы люди не покупали товары, а приобретали их в магазинах. Пример тому всем известный магазин Amazon. Любому интернет пользователю не составит труда найти в нём всё, что захочет.

[API](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) (от англ. application programming interface) – это интерфейс взаимодействия между сайтом и сторонними программами и серверами.

По сути API является программным посредником, который позволяет двум приложениям свободно взаимодействовать друг с другом. Сайты со спортивными событиями используют API сторонних сервисов, для того, чтобы показывать спортивные события в линии и в прямом эфире.

**ЦЕЛЬ КУРСОВОЙ РАБОТЫ**

Целью курсовой работы является закрепление практических навыков по Учебному плану «Бизнес-информатики», которые мы изучили за весь 3 курс.

1. Разработка интернет-приложений.
2. Конструирование программного обеспечения.
3. Базы данных.
4. Проектирование и оптимизация баз данных.
5. Реализация и тестирование приложений баз данных.

Спорт – это жизнь. Это давно всем известно. Все люди должны любить спорт. Неважно каким видом спорта увлекается человек. Помимо самого занятия спортом, люди также смотрят спортивные матчи. Такие как: футбол, баскетбол, смешанные единоборства.

Моя работа заключается в разработке сайта для просмотра спортивных событий с использованием современных технологий и работы с API, которые мне по просьбе на различных сайтах владельцы предоставляли бесплатно.

Технологии, выбранные мной:

* OpenServer. Локальный веб-сервер для виндовс
* PHP7. Язык программирования
* phpMyAdmin. База данных
* RedBeanPHP. **ORM** система
* CSS. Таблица каскадных стилей

Для выполнения данной работы не была выбрана ни одна IDE. Вся работа была выполнена на sublime text 3.

Sublime text 3 – это очень удобный текстовый редактор, который имеет огромный функционал, который другие текстовые редакторы предложить не могут. Данный редактор разрешает компилировать и интерпретировать код разных языков программирования.

Возможности:

- Удобная панель навигации.

- Палитра.

- API плагины на python2.x и python3.x

- Разделение экрана на 2, а то более частей.

**ОПИСАНИЕ ПРОГРАММЫ**

Модели:

* Модель главной страницы.
* Модель просмотра спортивных событий
* Модель авторизации и регистрации

В качестве интернет ресурса API использован ни один сайт.

Такие популярные сайты со спорт событиями как:

Sportradar.com

Api-football.com

football-data.org

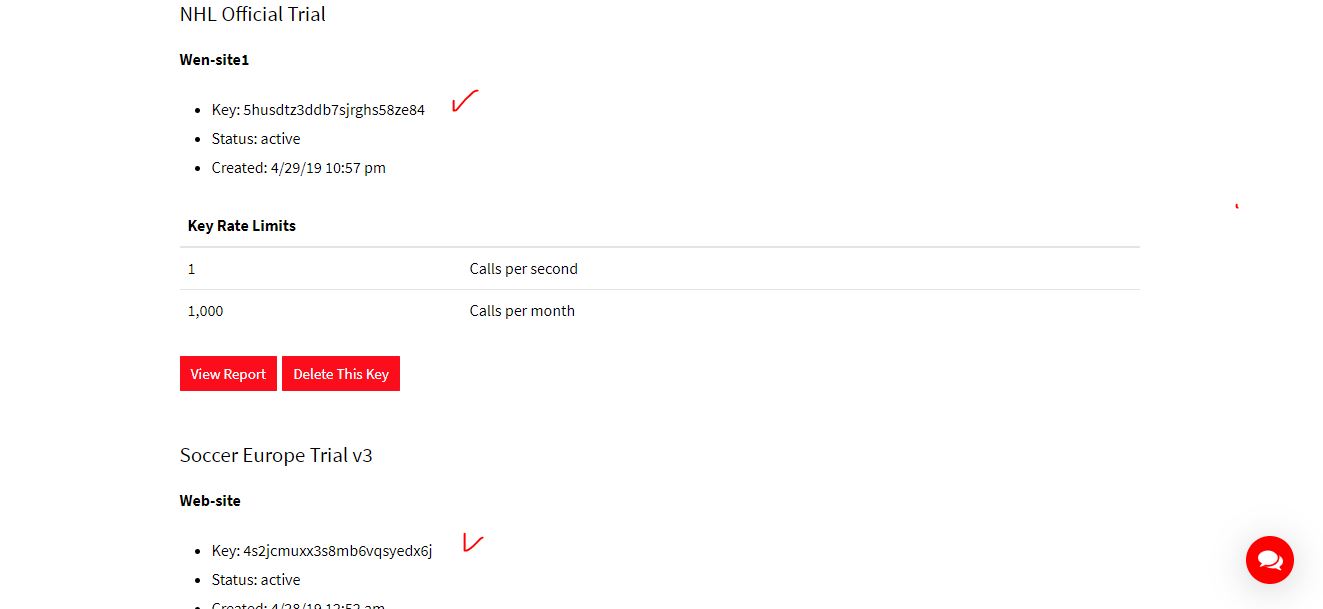
patreon.com

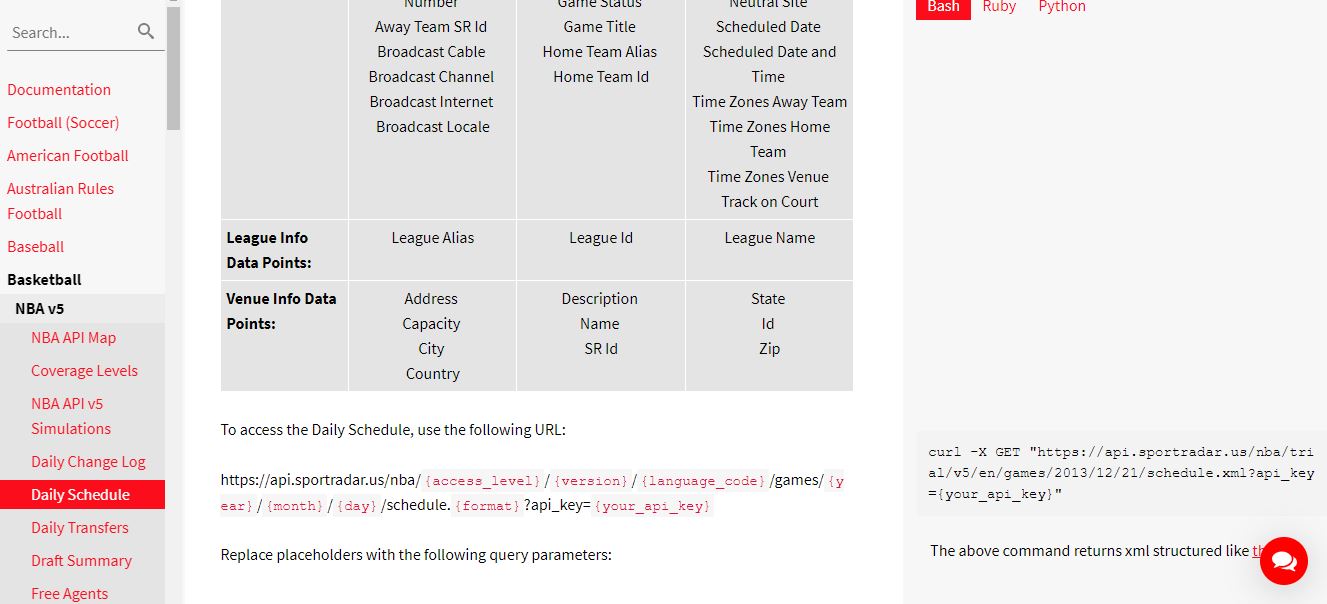
mysportsfeeds.com

Данные интернет ресурсы предоставляют платно или бесплатно API ключи, с помощью которых можно установить взаимодействие между своим и сторонним сайтом. Пользователь может просматривать различные спортивные события в формате Json или XML.

Для этого нужно зарегистрироваться на сайте, выбрать предпочитаемый вид спорта(также можно выбрать большое количество спортивных событий).

Подтвердить свой выбор с помощью электронной почты и свободно пользоваться предоставленными API ключами.

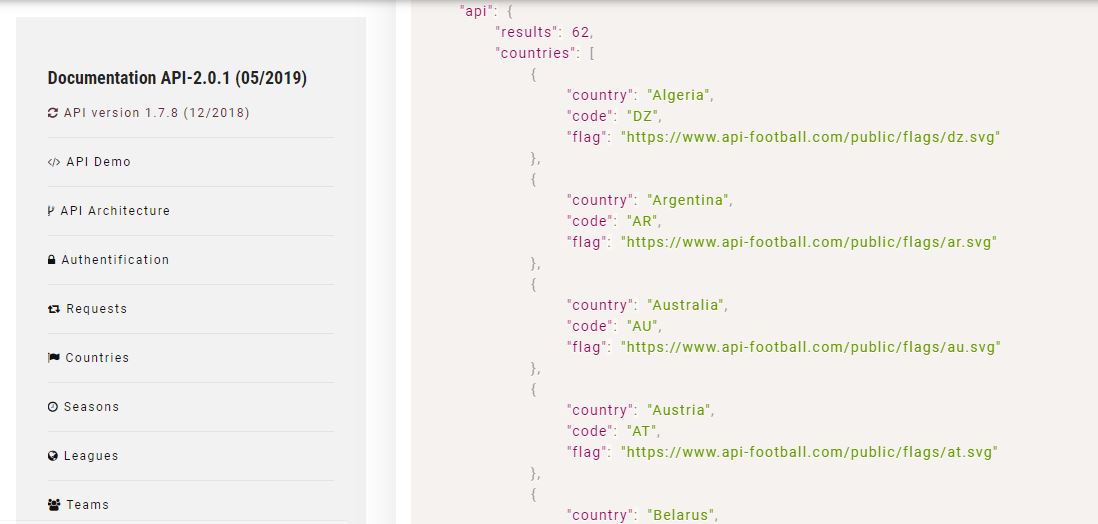


Документация сайта

Примеры API запросов:



Также документация, но уже с другого сайта:



**Архитектура приложения**

Основная панель навигации:

****

Панель навигации футбола:



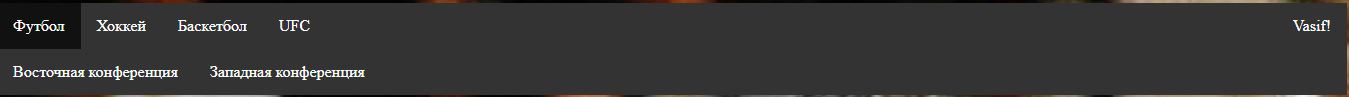
Состоит из личных встреч лондонских команд Арсенала и Челси.

Панель навигации хоккея:



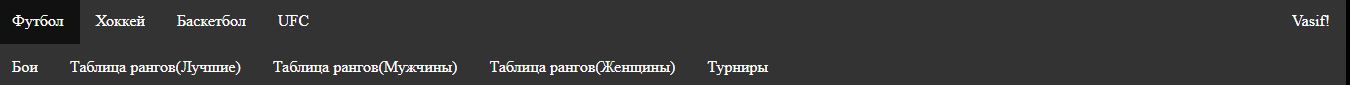
Состоит из новых трансферов команд, известной команды Team Kings, Восточной и Западной конференции.

Панель навигации баскетбола:



Состоит из команд Восточной и Западной конференции.

Панель навигации смешанных единоборств:



Состоит из проведённых недавних боёв, таблицы лучших бойцов независимо

от весовой категории и пола, таблица лучших бойцов мужского пола, женщин и турниров ММА.

**РАЗРАБОТКА ПРОГРАММЫ**

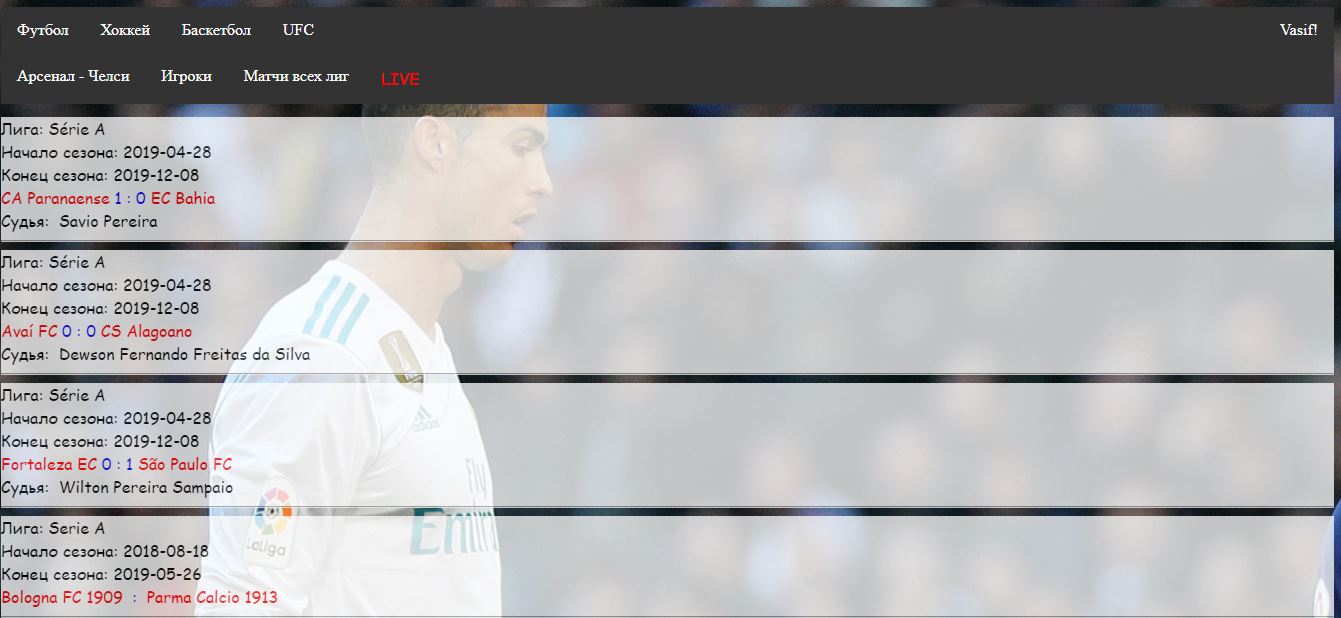
Для работы проекта требуется установка данного ПО:

* OpenServer
* Необходимый минимум системных ресурсов: 200 Мб RAM и 1 Гб на HDD;
* Windows (32-bit или 64-bit): Windows 8 / Windows 7 / Windows Server 2008 / Windows Vista / Windows XP SP3;
* Установленный набор библиотек Microsoft Visual C++ 2005-2008-2010 Redistributable Package x86;

Настройка, подключение и запросы к API:

Первым шагом курсовой работы было создание основной страны index.php

В начале я подключил файл ‘db.php’ в нём у меня создаётся подключение к базе данных через готовую библиотеку RedBean. О ней будет рассказано более подробно, когда будет описана реализация авторизации и регистрации на сайте.



Сама веб-страница.

В теге <head> Указав кодировку utf-8 и подключив файл стилей style.css, где будут прописаны все стили, следом я подключил с помощью функции ‘include()’

Шапку сайта – ‘header.php’.

Далее была введена суперглобальная переменная GET.

Ведена она была для подргузки всех страниц с разными видами спорта, которые находятся в папке temp.

Таких как – футбол, хоккей, баскетбол и смешанных единоборств UFC.



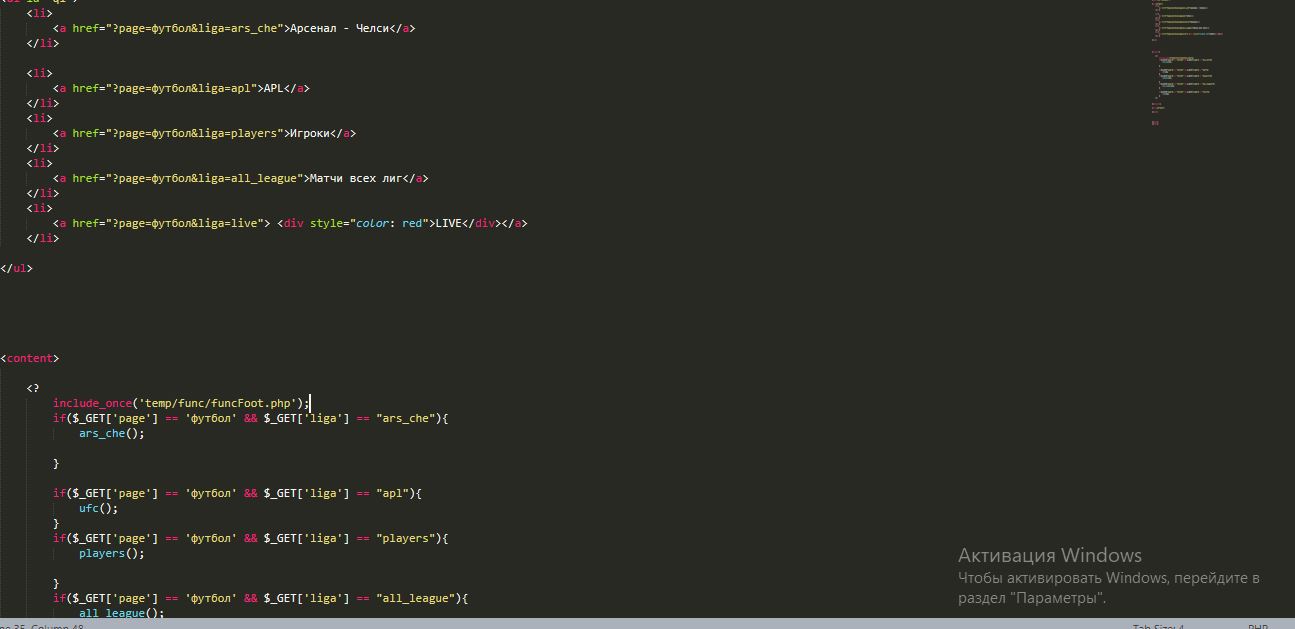
Первая страница которую я начал оформлять была страница с данными о футболе.

Указав вначале ссылки на страницы, ранее сказанным методом суперглобальной переменной GET, сделал список интересных для пользователей разделы.

Разделы содержать в себе - личные встречи лондонских команд (Арсенал-Чесли), состав команды, которая последний раз выиграла престижный турнир – Лигу Чемпионов, топ 5 игроков, которые играт на данный момент, матчи всех лиг, которые проходят сегодня(сыгранные и ещё не сыгранные матчи) и матчи, которые проходят в реальном времени.

Футбол

Далее я вставил теги <content> в которых создал условие. Если в адресной строке указан вид спорта ‘футбол’, а переменной ‘liga’ players, то будет подключена функция players(); Аналогично для всех других разделов.



Подробнее о подключенных функциях:

Я создал отдельную папку func в которой также создал отдельно для каждого вида спорта свой файл. Для футбола был создан файл funcFoot.php

Так как я уже указал ранее подключённую функцию players(); буду описывать именно её.

Я зашёл в интернет и ввёл в поиске бесплатные API football запросы.

Бесплатных в открытом доступе API запросы найти сложно. Обычно они все стоят денег и требуют подписку на сайт. Я связывался с администрацией сайтов, чтобы мне, как студенту предоставили бесплатные API ключи.

Несколько администраторов любезно предоставили мне API ключи.

Первый мой API запрос был построен данным запросом:

$uri = ‘https://api.sportradar.us/soccer-t3/eu/en/players/sr:player:12994/profile.json?api\_key=4s2jcmuxx3s8mb6vqsyedx6j’;

$reqPrefs['http']['method'] = 'GET';

$reqPrefs['http']['header'] = 'X-Auth-Token: a41319aa40164131af24b45c8afe216c';

$stream\_context = stream\_context\_create($reqPrefs);

$response = file\_get\_contents($uri, false, $stream\_context);

$matches = json\_decode($response, true);

echo "<pre>";

print\_r($matches);

echo "<pre>";

В ходе работы я совершенствовал свой код, создавал более красивые и конструированные запросы.

В данной функции был создан массив с игроками

$array = array('sr:player:12994', 'sr:player:750', 'sr:player:15466', 'sr:player:256219', 'sr:player:1');

ID игроков я узнал благодаря быстрого чата англоязычного сайта sportradar.com

Там мне подробно указали где найти ID лиг, команд и игроков

В самом запросе я вывел позицию, национальность, рост, вес, ведущую ногу, текущий клуб, сборную за котороую выступает футболист, количество проведённых матчей, количество голов и количество голевых передач игрока.

Вывод данных значений было осуществлено с помощью цикла foreach.

Фунция players();

function players(){

$array = array('sr:player:12994', 'sr:player:750', 'sr:player:15466', 'sr:player:256219', 'sr:player:1');

for($j = 0; $j < 5; $j++){

$uri = 'https://api.sportradar.us/soccer-t3/eu/en/players/'.$array[$j].'/profile.json?api\_key=4s2jcmuxx3s8mb6vqsyedx6j';

$reqPrefs['http']['method'] = 'GET';

$reqPrefs['http']['header'] = 'X-Auth-Token: a41319aa40164131af24b45c8afe216c';

$stream\_context = stream\_context\_create($reqPrefs);

$response = file\_get\_contents($uri, false, $stream\_context);

$matches = json\_decode($response, true);

echo "<pre>";

//print\_r($matches);

echo "<pre>";

echo '<div style="font-family: Lobster, cursive; background: white; opacity: 0.75; ">'

. "Имя: "

. '<div style="color:red; font: 200%;">' . $matches['player']['name'] . '</div>'

. "<br>Позиция: " . $matches['player']['type']

. "<br>Национальность: " . $matches['player']['nationality']

. "<br>Рост: " . $matches['player']['height']

. "<br>Вес: " . $matches['player']['weight']

. "<br>Ведущая нога: " . $matches['player']['preferred\_foot']

. "<br>Клуб: " . $matches['teams']['0']['name']

. "<br>Сборная: " . $matches['teams']['1']['name']

. "<br>Игр сыграно: " . $matches['statistics']['totals']['matches\_played']

. "<br>Голы: " . $matches['statistics']['totals']['goals\_scored']

. "<br>Голевые передачи: " . $matches['statistics']['totals']['assists']

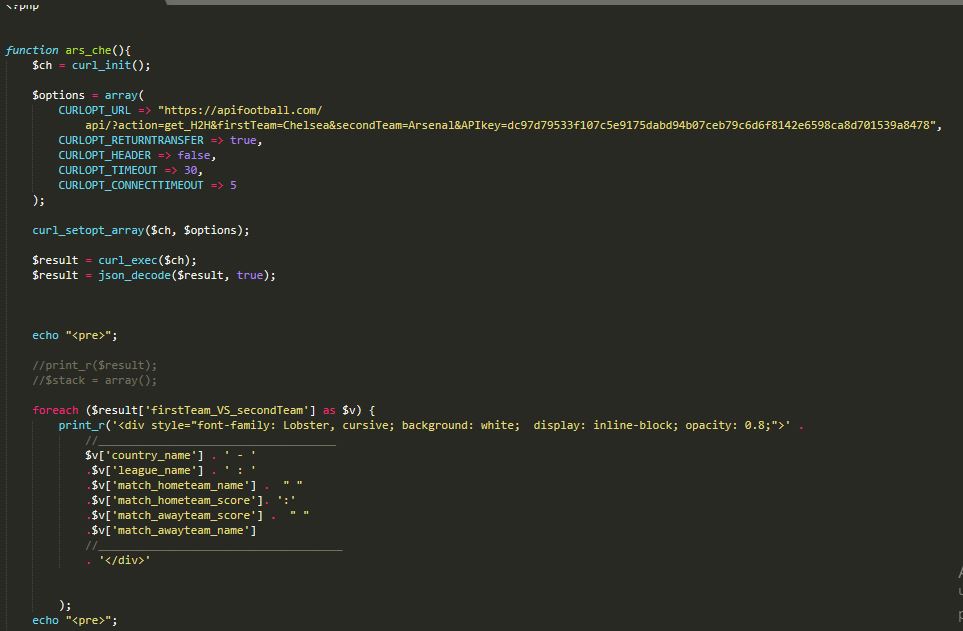
. '</div>';

}

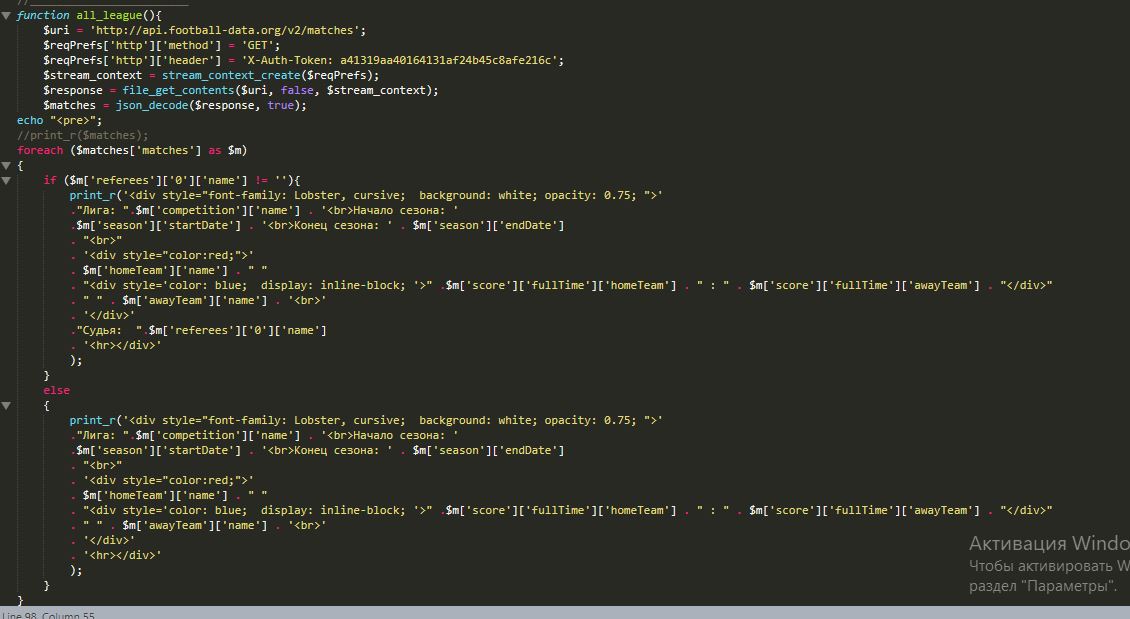
}

Аналогичным способом я вывел и все остальные данные функций, которые относились к футболу.

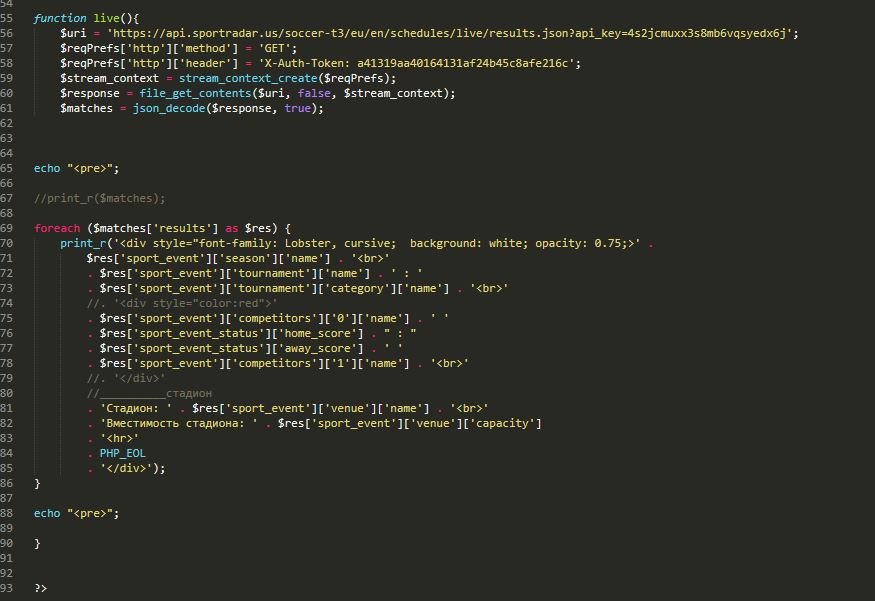
Функция личных встреч лондонских команд Арсенал – Челси are\_che();



Функция матчей всех лиг all\_leagues();



Фунция лайв игр live();



На этом раздел “Футбол” закончен.

Следующий раздел – это раздел “Хоккей”.

Хоккей

На странице хоккея я также создал список с трансферами, клубом – Team Kings, Восточная и Западная конференция лиги NHL.

Через теги <content> как и в случае с футболом были указаны через глобальную переменную GET, страницу – ‘page’ и лигу – ‘leagues’.

Дословно – если в глобальной переменной ‘page’ указан ''хоккей” а в “liga” трансферы, до будет подключена функция transfers();

Подробнее о функции transfers();

В ранее указанной папке func лежит файл funcHoc.php указывать на это больше не будем.

Здесь я использовал запрос другого типа, но ответ выдавался также в формате JSON.

В данной функции я вывел имя игрока, позицию, номер и переход из одного клуба в другой.

function transfers(){

$ch = curl\_init();

$options = array(

CURLOPT\_URL => "https://api.sportradar.us/nhl/trial/v5/en/league/2015/09/21/transfers.json?api\_key=5husdtz3ddb7sjrghs58ze84",

CURLOPT\_RETURNTRANSFER => true,

CURLOPT\_HEADER => false,

CURLOPT\_TIMEOUT => 30,

CURLOPT\_CONNECTTIMEOUT => 5

);

curl\_setopt\_array($ch, $options);

$result = curl\_exec($ch);

$result = json\_decode($result, true);

echo "<pre>";

//print\_r($result);

foreach ($result['players'] as $players) {

print\_r('<div id = "oroz">' .

$players['full\_name'] . "<br>Position: " . $players['primary\_position'] . '<br>' . "Number: " . $players['jersey\_number'] . "<br>Moved from " . $players['transfers']['0']['from\_team']['name'] . " to " . $players['transfers']['0']['from\_team']['market'] . "<hr>" . '</div>' . PHP\_EOL);

}

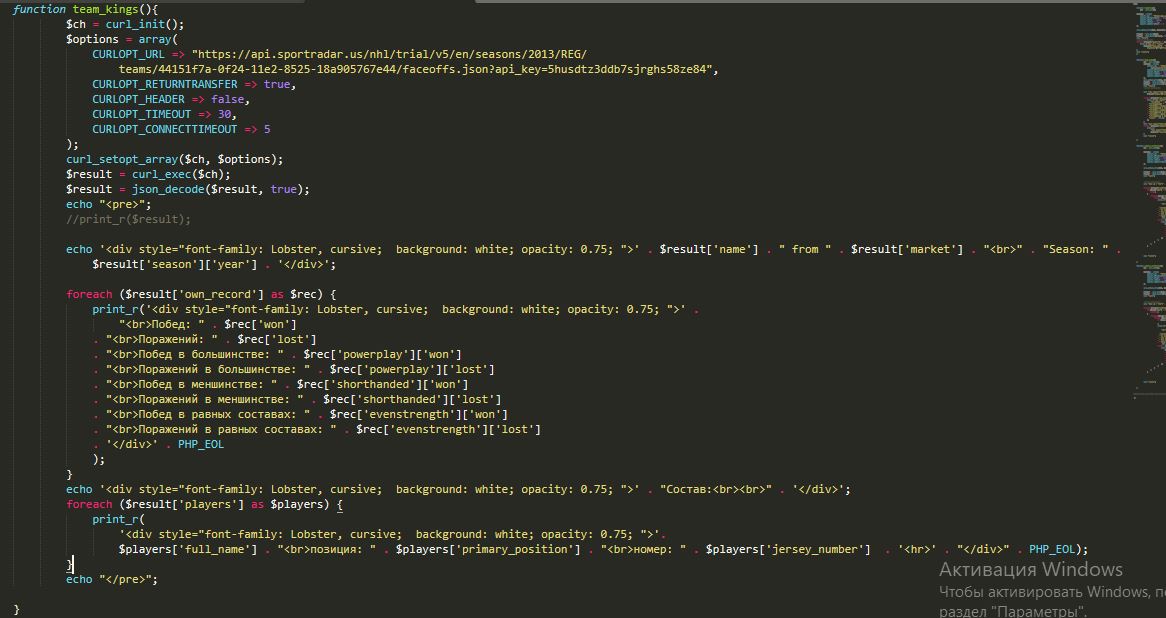
echo "</pre>";

}

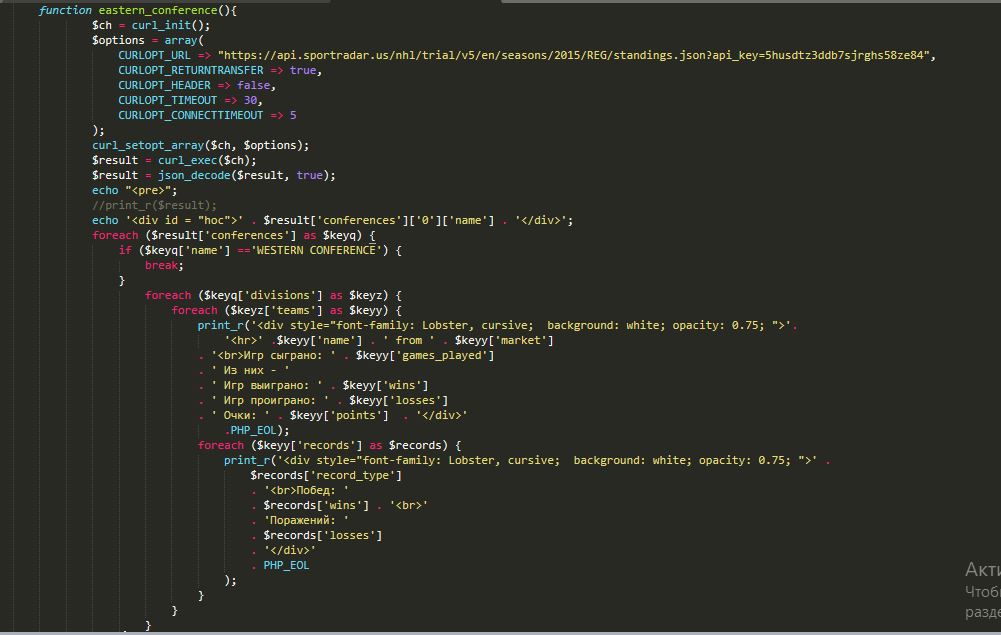
В ходе был использован ранее упомянутый цикл foreach.

Аналогично для других функций хоккея.

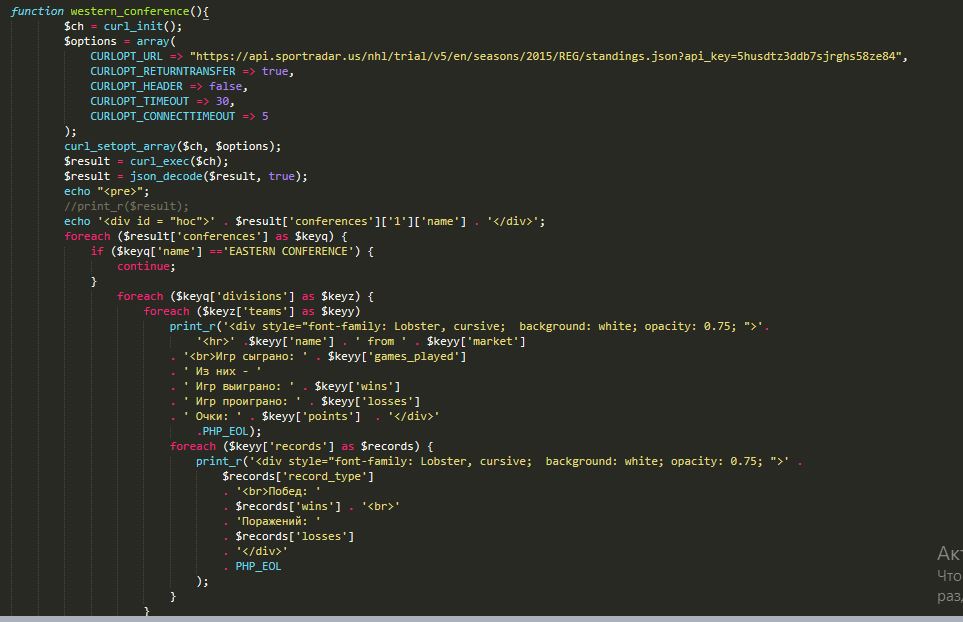
Функция team\_kings();



Функция eastern\_conference();



Функция western\_conference();



Баскетбол:

На странице баскетбола я создал разделы Западная и Восточная конференция высшей лиги NBA.

Так как запрос на Восточную и Западную конференцию один и тот же, было создано условие:

if ($conf['name'] =='WESTERN CONFERENCE') {

break;

}

Цикл заканчивается, когда $conf[‘name’] равняется 'WESTERN CONFERENCE'

В восточной конференции через цикл foreach вытаскиваю данные о имени команды и за какой город выступает команда. Также вытаскивается адрес стадиона.

function EASTERN\_CONFERENCE(){

$ch = curl\_init();

$options = array(

CURLOPT\_URL => "https://api.sportradar.us/nba/trial/v5/en/league/hierarchy.json?api\_key=q93as7v85zkke6r479ey9wkn",

CURLOPT\_RETURNTRANSFER => true,

CURLOPT\_HEADER => false,

CURLOPT\_TIMEOUT => 30,

CURLOPT\_CONNECTTIMEOUT => 5

);

curl\_setopt\_array($ch, $options);

$result = curl\_exec($ch);

$result = json\_decode($result, true);

echo "<pre>";

//print\_r($result);

foreach ($result['conferences'] as $conf) {

if ($conf['name'] =='WESTERN CONFERENCE') {

break;

}

foreach ($conf['divisions'] as $div) {

foreach ($div['teams'] as $teams) {

print\_r('<div style="font-family: Lobster, cursive; background: white; opacity: 0.75; ">'

. '<div style="color: red">'

. $teams['name'] . ' from ' . $teams['market']

. '</div>'

. '<br>Город: ' . $teams['venue']['city']

. '<br>Адрес: ' . $teams['venue']['address'] . '<hr>'

. '</div>'

. PHP\_EOL);

}

}

}

echo "</pre>";

}

В западной конференции код один и тот же, только выполняется проверка на восточную конференцию, если цикл видит, что ($conf['name'] =='EASTERN CONFERENCE'), то действие пропускается с помощью ‘continue’.

if ($conf['name'] =='EASTERN CONFERENCE') {

continue;

}

Функция WESTERN\_CONFERENCE();



UFC

В смешанных единоборствах были созданы разделы такие как – бои, таблица рангов(лучшие), таблица рангов(мужчины), таблица рангов(женщины), турниры

Первый раздел – это раздел бои. Здесь указаны последние бои смешанных единоборств UFC.

В запросе функции figth выводится название спорта, категория, место проведения боя, имена спортсменов, весовая категория и детали матча, такие как победитель и продолжительность боя.

Реализовано это было также с помощью foreach

foreach ($result['summaries'] as $key) {

if ($key['sport\_event\_status']['winner'] == 'home\_team') {

# code...

print\_r('<div style="font-family: Lobster, cursive; background: white; display: inline-block; opacity: 0.8;">'

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

. 'Спорт: ' . $key['sport\_event']['sport\_event\_context']['sport']['name']

. " Категория: " .

$key['sport\_event']['sport\_event\_context']['category']['name']

. "<br>Место проведения: " .

$key['sport\_event']['sport\_event\_context']['season']['name'] . '<br>'

. '<div style="color:red"><strong><i><hr>'

. $key['sport\_event']['competitors']['0']['name']

. ' vs ' . $key['sport\_event']['competitors']['1']['name']

. '</i></strong></div>'

. "<br>Весовая категория: " . $key['sport\_event\_status']['weight\_class']

. "<br>Количество раундов: " . $key['sport\_event\_status']['final\_round']

. "<br>Продолжительность боя: " . $key['sport\_event\_status']['final\_round\_length']

. '<br>Победитель: '

. $key['sport\_event']['competitors']['0']['name']

. '<hr>'

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

. '</div>'

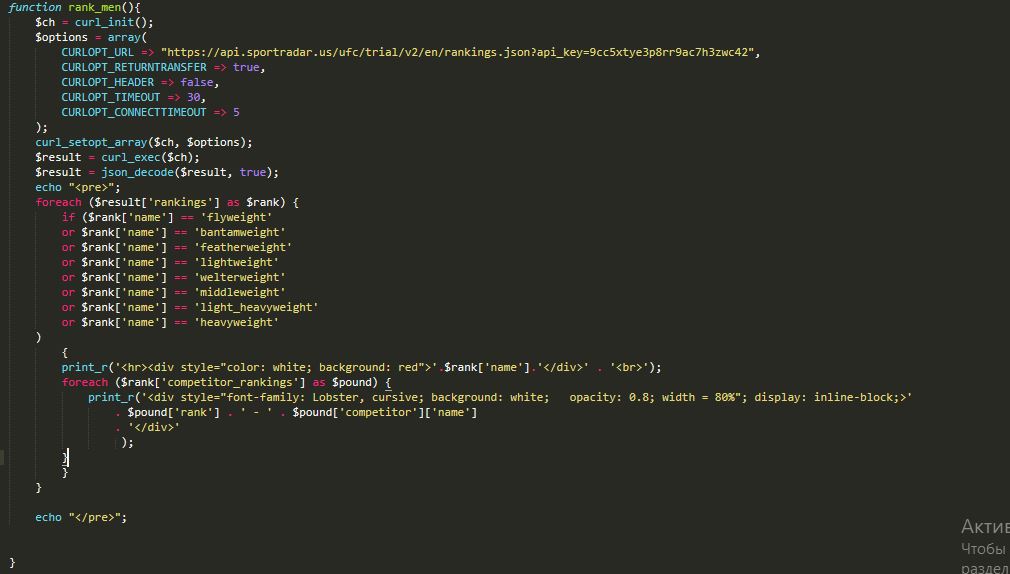
. PHP\_EOL

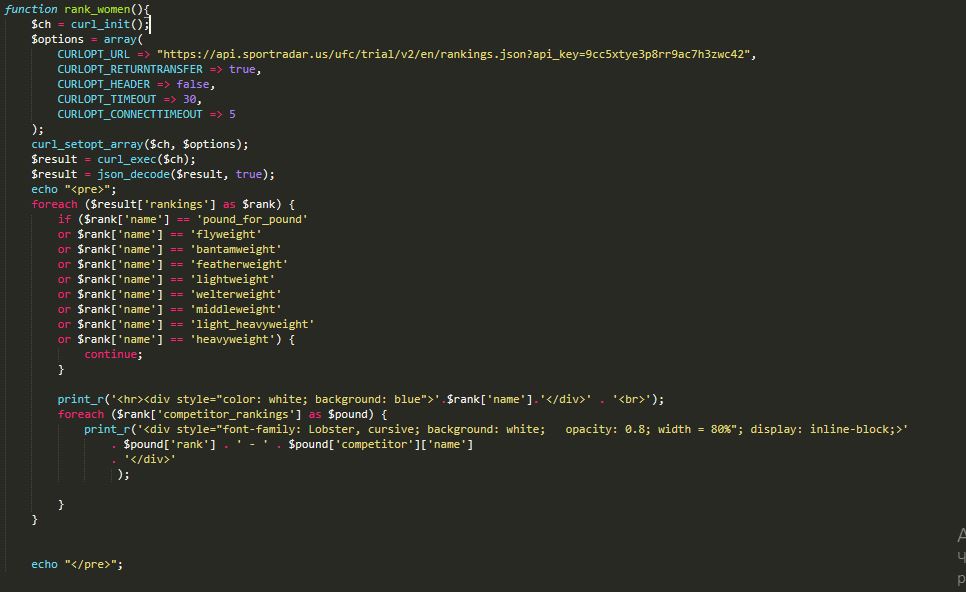
);

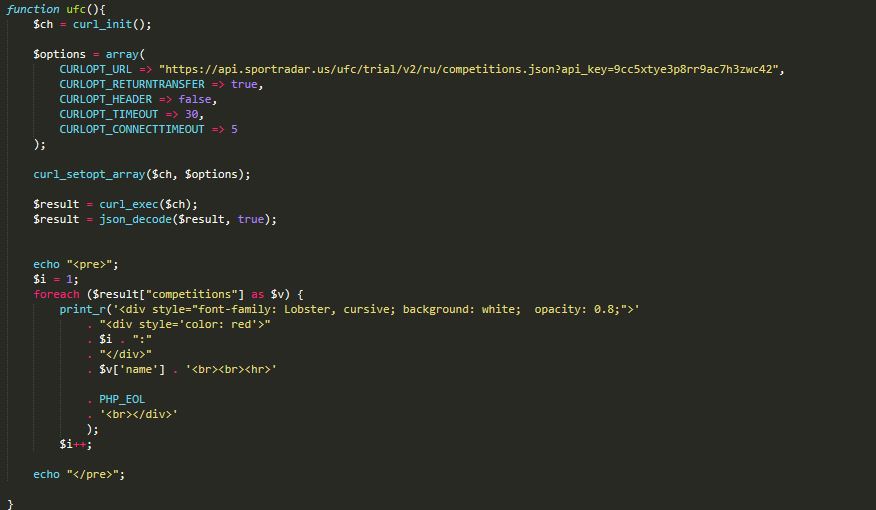
}

В функциях rankBEST, rank\_men, rank\_women, ufc выведены лучше бойцы вне зависимости от весовой категории, ранги мужчин по весовым категориям, ранги женщин по весовым категориям, таблица турниров соответственно.

Функции rankBEST();, rank\_men();, rank\_women();, ufc();







Регистрация и Авторизация

Регистрация и авторизация реализована с помощью готовой библиотеки RedBeanPHP.

RedBeanPHP - это простая в использовании **ORM** для **PHP**. Это **библиотека Zero Config** ORM, которая *«автоматически»* создает **схему** вашей базы **данных**.

Файлы с регистрации и авторизации:

db.php, index.php, login.php, login.php, logout.php, rb.php, signup.php.

Подробно о каждом файле.

В начале каждого файла мы подключаем ‘db.php’.

db.php:

В данной файле мы подключаем ‘rb.php’, где прописаны готовые библиотеки.

Также c помощью встроенной функции библиотеки RedBean происходит подключение к базе данных

(

R::setup('mysql:host=localhost;dbname=course', 'root', '');

session\_start();

)

index.php:

На данной странице у пользователя есть выбор между авторизацией и регистрацией на сайте. Сессия начинается в данном блоке.

signup.php:

Регистрация происходит через отправку заполненной пользователем формы.

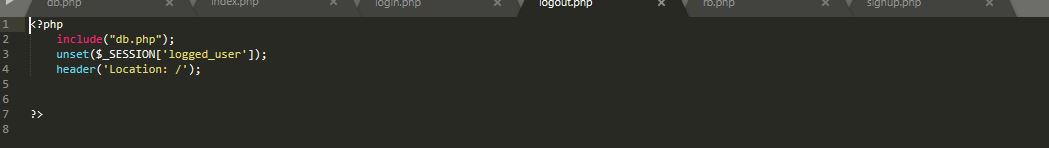
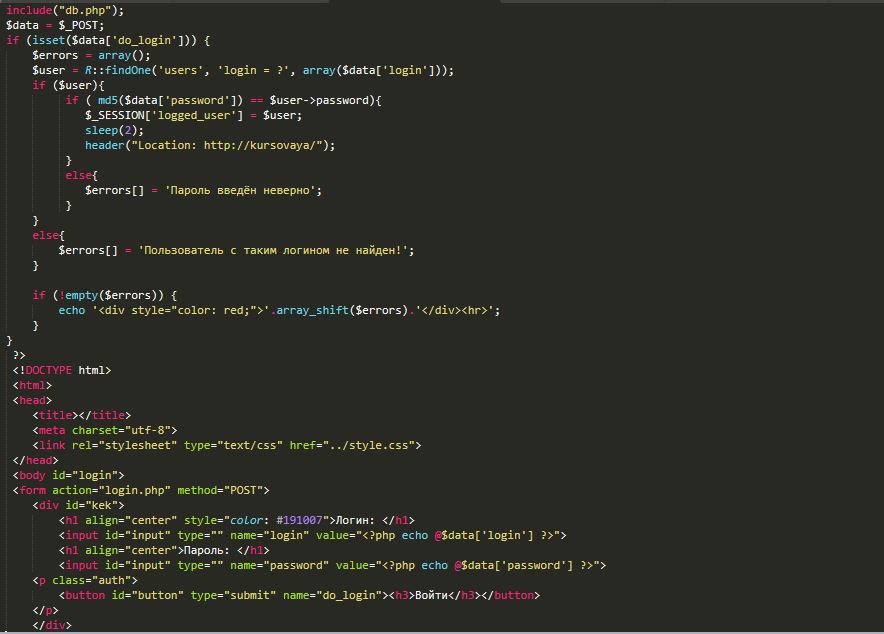
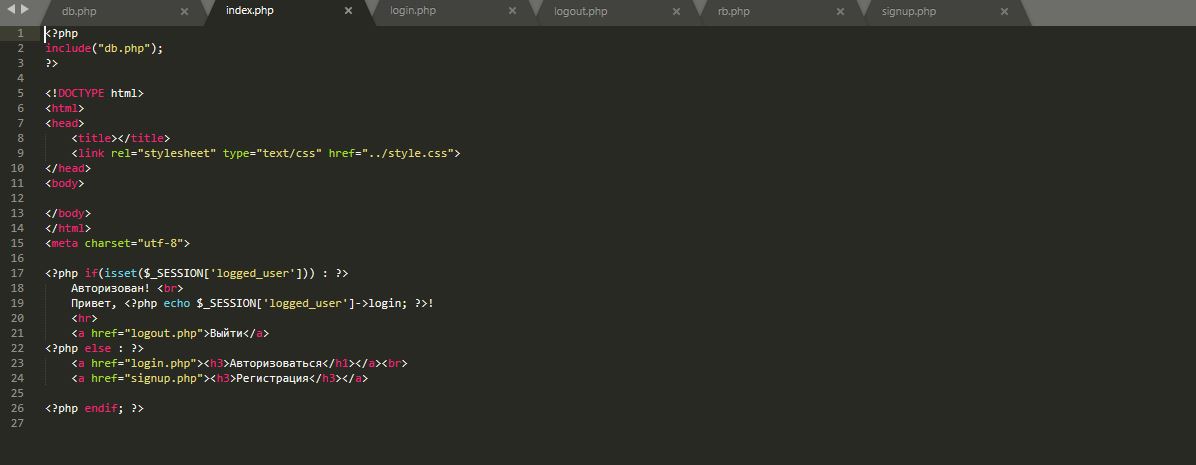
RedBean автоматически создаёт таблицу в базе данных.

login.php:

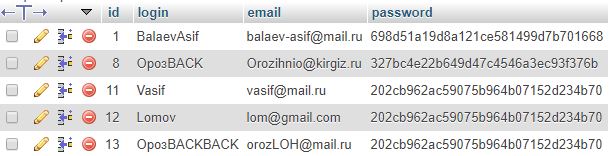
Авторизация производится через проверку функцией ‘isset()’, если RedBean находит в базе существующий логин и проходит проверку вместе с паролем, то сессия записывает данного пользователя.

logout.php:

Здесь закрывается и заканчивается сессия. Пользователь выходит из своего аккаунта.



База данных пользователей:



Заключение:

По итогам курсовой работы был создан сайт спортивных событий.

В ходе работы были изучены навыки работы с API, для работы и взаимодействием с другими сайтами.

Также была изучена готовая библиотека RedBeanPHP, которая имеет большое количество плюсов. Такие как автоматическая защита от sql-инъекций.

В ходе работы была создана база данных с помощью RedBeanPHP.

Сайт выполняет свой функционал и показывает матчи разных видов спорта. Большим плюсом можно отметить то, что была выполнена реализация футбольных матчей в прямом эфире.

Сайт полностью протестирован.

В дальнейшем сайт будет обслуживаться поддержкой и расширением функционала.

Список литературы:

<https://allsportsapi.com/basketball-api-documentation>

<https://developer.sportradar.com/>

<https://www.football-data.org/documentation/quickstart>

<https://www.patreon.com/mysportsfeeds/overview>

<https://www.mysportsfeeds.com/browse/>

<https://www.youtube.com/>

<https://redbeanphp.com/>

[http://www.cyberforum.ru](http://www.cyberforum.ru/)

[https://habr.com](https://habr.com/)

[http://stackoverflow.com](http://stackoverflow.com/)