МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий   
Кафедра Информационных систем и технологий   
Специальность 1-98 01 03 Программное обеспечение информационной безопасности мобильных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

**«Разработка web-сайта “Фан сайт футбольного клуба Liverpool**”

Выполнил студент Сероокий Никита Дмитриевич

(Ф.И.О.)

Руководитель проекта асс. Нистюк О.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., доц. Смелов В.В .

(учен. степень, звание, должность, Ф.И.О., подпись)

Консультант асс. Нистюк О.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер асс. Нистюк О.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовая работа защищена с оценкой

**Содержание**

[Введение 3](#_Toc27003865)

[1. Общие сведения 4](#_Toc27003866)

[1.1. Постановка задачи 4](#_Toc27003867)

[1.2. Обзор программных средств 4](#_Toc27003868)

[2. Разработка веб-сайта 6](#_Toc27003869)

[2.1. Разработка содержания и структуры веб-сайта 6](#_Toc27003870)

[2.2. Ход работы веб-сайта 6](#_Toc27003871)

[2.2.1. Главная страница 9](#_Toc27003872)

[2.2.2. Страница «Состав» 10](#_Toc27003873)

[2.2.3. Страница «История» 14](#_Toc27003874)

[2.2.4. Страница «О сайте» 15](#_Toc27003875)

[2.2.5. Страница «Игрок» 15](#_Toc27003876)

[3. Заключение 17](#_Toc27003877)

[4. Список источников 18](#_Toc27003878)

[5. Приложение 19](#_Toc27003879)

[Приложение А 19](#_Toc27003880)

[Приложение Б 24](#_Toc27003881)

[Приложение В 37](#_Toc27003882)

# **Введение**

В настоящее время сеть Интернет стала неотъемлемой частью жизни современного человека, так как является самым объемным и быстрым источником информации. На основе Интернета работает Всемирная паутина.

Всемирная паутина (World Wide Web) – это распределенная система доступа к гипертекстовым документам, существующая в сети Internet. Всемирная паутина вызвала революцию в сфере информационных технологий. Совокупность логически связанных между собой веб-страниц (гипертекстовых документов, размещенных в World Wide Web) образуют веб-сайт.

Большая часть информации, размещенной во Всемирной паутине, представляет собой гипертекст. Гипертекст – это система страниц с текстом, связанных ссылками. Для создания, хранения и отображения гипертекста в веб-пространстве используют HTML – язык гипертекстовой разметки. В настоящее время актуальной версией языка является HTML5.

Для улучшения зрительного восприятия веб-страниц используются каскадные таблицы стилей CSS. При помощи этой технологии появилась возможность создавать единые стили для множества веб-страниц. Также на страницах широко применяется язык программирования JavaScript для придания интерактивности страницам.

Тема курсовой работы – «разработка веб-сайта “Фан сайт футбольного клуба Liverpool”».

Цель курсовой работы: разработать состав и структуру веб-сайта на тему «Музеи Минска», наполнить веб-страницы тематическим контентом, а также создать удобный пользовательский интерфейс.

# **Общие сведения**

## **Постановка задачи**

Веб-сайт, разработанный на тему «Фан сайт футбольного клуба Liverpool», должен соответствовать следующим требованиям:

* Предоставление информации о игроках и истории футбольного клуба “Ливерпуль”. Это может быть достигнуто путем разработки специальной страницы-каталога, на которой игроки будут располагаться в виде списка с краткой информацией. Сами же страницы игроков должны размещать описание игрока, картинку, а также такие поля, как игровой номер, гражданство, инстаграм и возраст игрока.
* Возможность поиска игроков в каталоге по фамилии, а также с применением фильтров по позиции, по состоянию здоровья.
* Сайт должен быть разработан при помощи блочной верстки с использованием стилей CSS. Также важным требованием является корректное отображение сайта в современных браузерах и наличие как десктопной версии, так и мобильной.
* Все страницы данного веб-сайта должны содержать такие базовые элементы пользовательского интерфейса, как:
  + Шапка (header) – элемент, содержащий логотип и название веб-сайта, основное меню, а также форму поиска;
  + Основная часть сайта с контентом и «хлебными крошками»;
  + Подвал (footer) – элемент, содержащий таблицу с информацией о разработчике, а также счетчик посетителей сайта.
* Веб-сайт должен быть размещен на хостинге в сети Internet.

## **Обзор программных средств**

В ходе разработки веб-сайта использовался текстовый редактор Notepad++. Notepad++ – свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса, разметки, а также языков описания аппаратуры VHDL и Verilog. Еще данный редактор использовался для работы с XML-файлами в процессе разработки веб-сайта.

Также при разработке использовались средства отладки браузера Google Chrome 78. Chrome – это браузер, разрабатываемый компанией Google на основе свободного браузера Chromium и движка Blink. Особенностью данного браузера является то, что он ориентирован на повышение скорости, безопасности и стабильности. Google Chrome использует высокопроизводительный движок V8 для обработки JavaScript. Браузер имеет встроенный диспетчер задач, что позволяет завершать процессы вкладок и плагинов вручную.

Помимо Chrome, при разработке веб-сайт проверялся на работоспособность и в других браузерах, таких, как Mozilla Firefox 71, Microsoft Edge 44 и Internet Explorer 11.

За поддержкой браузерами различных элементов языков HTML, CSS и JavaScript помогал следить инструмент «Can I use». Данный ресурс предоставляет актуальные таблицы поддержки браузерами front-end технологий.

# **Разработка веб-сайта**

## **Разработка содержания и структуры веб-сайта**

Веб-сайт «Фан сайт футбольного клуба Liverpool» состоит из пяти веб-страниц:

* Главная страница;
* Состав;
* История;
* О сайте;
* Персональная страница игрока.

На главной странице сайта размещена общая информация о тематике сайта: футбольный клуб Ливерпуль, его достижения, соперники, положение на мировой футбольной арене, а также ссылка на генерального спонсора команды. В «подвале» страницы расположена информация об авторе сайта, которая хранится в файле student.xml.

На странице «Состав» располагаются игроки с кратким описанием. При нажатии на игрока в каталоге происходит переход на страницу игрока с подробным описанием. Вся информация о игроках структурированно хранится в файле players.xml.

На странице «История» размещена информация о создании, развитии, а также о трагедии клуба, связанной с гибелью большого количества фанатов.

На странице «О сайте» размещена информация о создании сайта.

Страница игрока, помимо текстовой информации о игроке, также содержит несколько графических изображений в галерее.

## **Ход работы веб-сайта**

Рассмотрим общие для каждой веб-страницы детали пользовательского интерфейса сайта «Фан сайт футбольного клуба Liverpool».

В верхней части страницы расположена «шапка», которая выделена на рисунке 1.

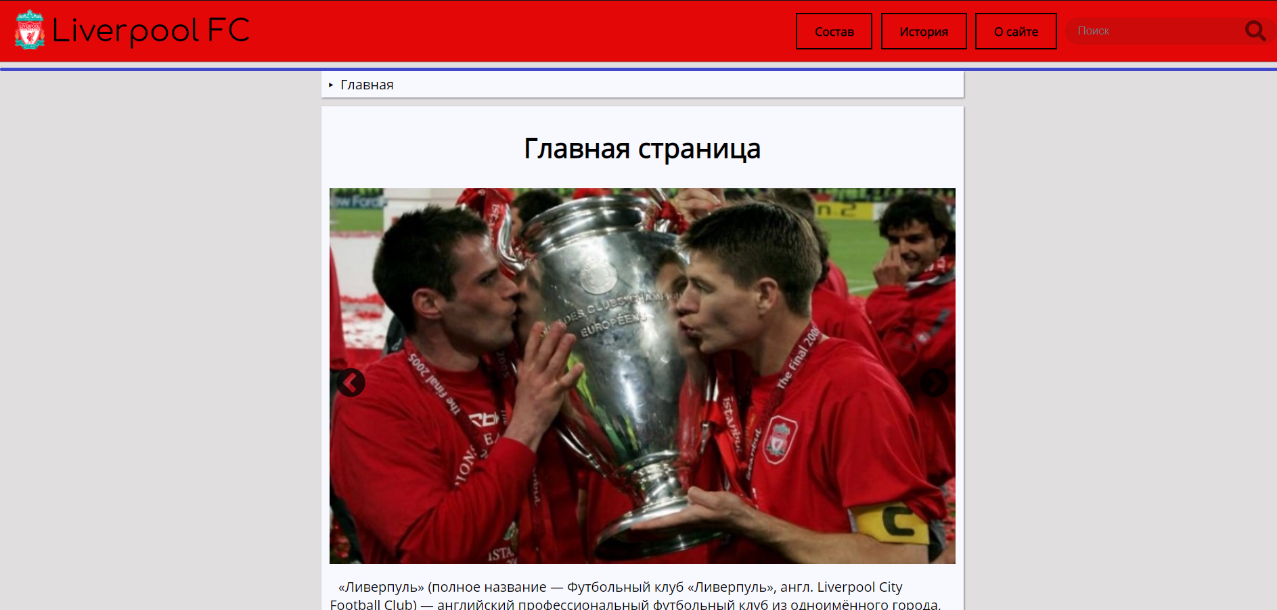


Рисунок 1 – Шапка веб-сайта

В левой части шапки расположены логотип и название сайта, в правой – меню и форма поиска.

Основное меню веб-сайта состоит из трех кнопок: «Состав», «История» и «О сайте». Данные кнопки являются гиперссылками на соответствующие веб-страницы сайта. При наведении курсора на любую из кнопок основного меню происходит анимация изменения цвета кнопки. При наведении курсора на кнопку «Состав» появляется выпадающее меню с возможностью быстро выбрать позицию интересующих игроков, для последующей работы с фильтрами на странице каталога (рисунок 2).

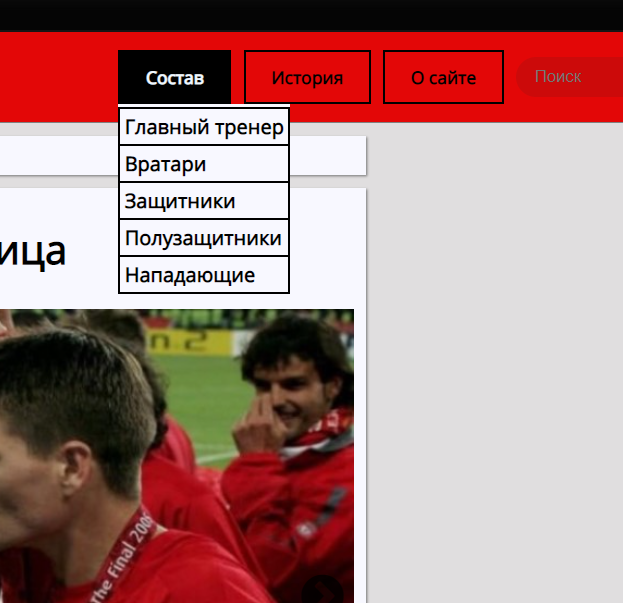


Рисунок 2 – выпадающее меню каталога

В версии для мобильных устройств меню и форма поиска расположены под шапкой, при этом кнопки меню расположены вертикально (рисунок 3).



Рисунок 3 – Шапка в мобильной версии сайта

Далее на каждой странице веб-сайта расположен элемент «хлебные крошки». В нем расположен путь от текущей страницы до главной в виде гиперссылок, при этом название текущей страницы гиперссылкой не является (рисунок 4).

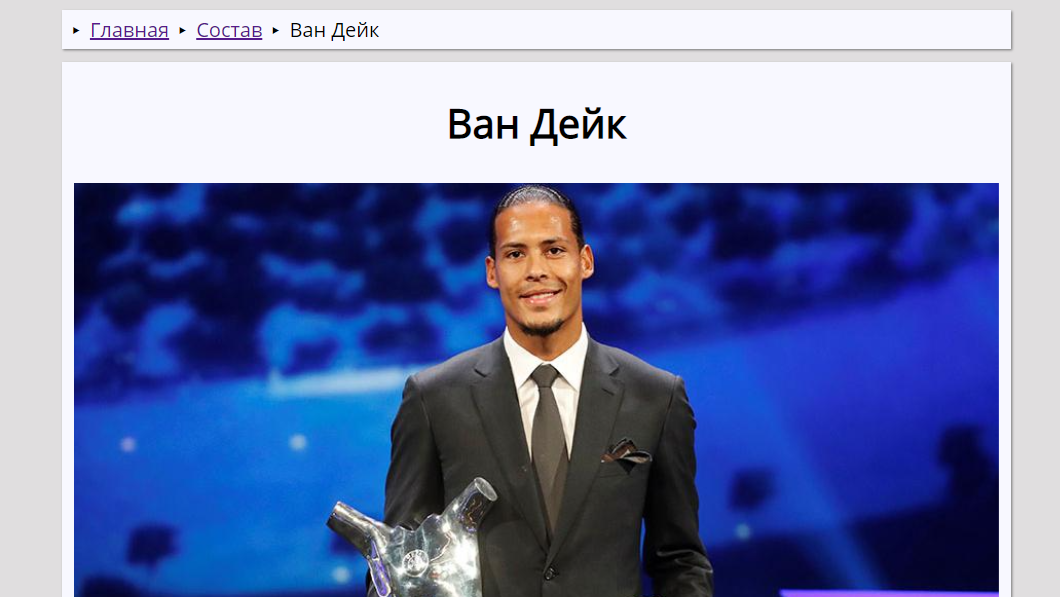


Рисунок 4 – Элемент «хлебные крошки» на примере страницы музея

После элемента «хлебные крошки» идет элемент, в котором размещается основной контент веб-страницы (для каждой страницы разный).

Внизу страницы располагается элемент «подвал». В нем размещен счетчик посетителей «Яндекс-Метрика», а также таблица сведений о создателе сайта на главной странице (рисунок 5).

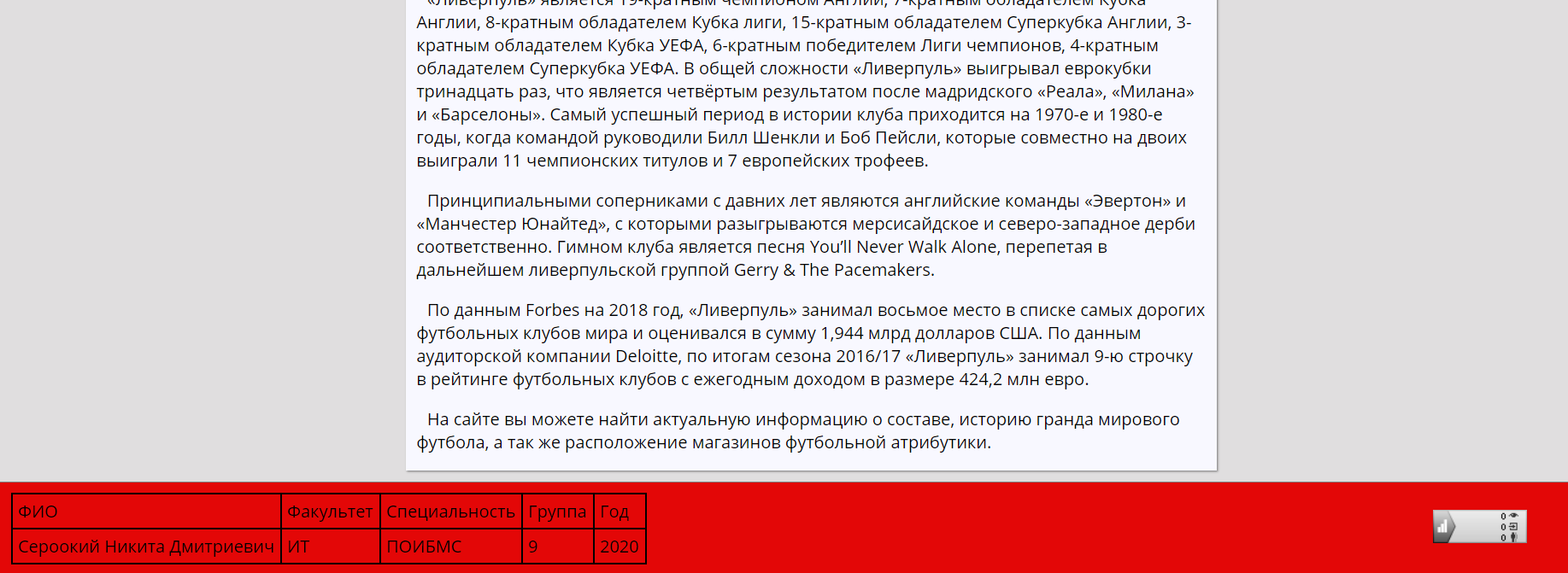


Рисунок 5 – «подвал» страницы

Теперь перейдем к особенностям отдельных страниц.

### **Главная страница**

Особенностью главной страницы веб-сайта является наличие элемента галереи-карусели (рисунок 6), в которой расположены два изображения вспомогательных страниц, а также изображение спонсора клуба. При нажатии на изображение происходит переход на вспомогательные страницы или сайт партнера.



Рисунок 6 – «Карусель» изображений на главной странице

Данная карусель выполнена на основе jQuery плагина slick. Прокрутка изображений происходит автоматически. При нажатии на кнопки «влево» и «вправо» можно управлять прокруткой самостоятельно.

### **Страница «Состав»**

Страница **«**Состав» является наиболее важной страницей веб-сайта «Фан сайт футбольного клуба Liverpool», так как основная часть контента сайта расположена именно здесь. Данная страница изображена на рисунке 7.

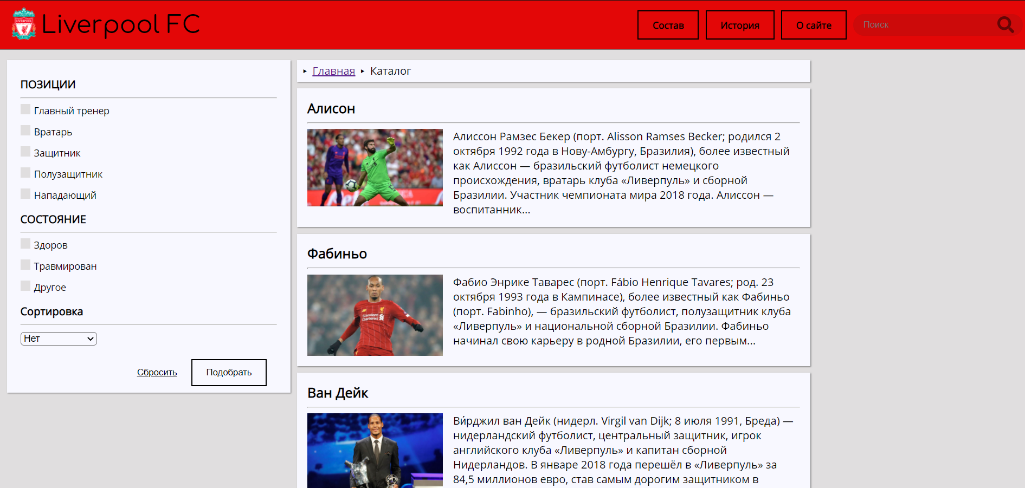


Рисунок 7 – страница «Состав»

Основной частью этой страницы является список игроков с краткой информацией в центре экрана. Данный список парсится из файла players.xml (см. Приложение В) при помощи скрипта catalog.js (см. Приложение Б) следующим образом:

Сначала отправляем запрос XMLHttpRequest() на сервер. Далее при помощи свойства responseText получаем содержимое xml-файла в виде строки. После этого парсим строку при помощи DOMParser и метода parseFromString(). Эти действия выполняет функция PlayerParser(), возвращающая DOM документ:

function PlayerParser() {

var xmlhttp;

if (window.XMLHttpRequest)

{ // для IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

}

else

{ // для IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.open("GET","https://raw.githubusercontent.com/SNikita24/Coursework/main/players.xml ",false);

xmlhttp.overrideMimeType('application/xml');

xmlhttp.send();

var xmlDoc = xmlhttp.responseText;

var parser = new DOMParser();

return parser.parseFromString(xmlDoc, "application/xml");

}

После выполнения данной функции преобразуем игроков из xml в объекты Player при помощи следующего конструктора:

function Player(title, picture, info, district, tag, cost) {

this.title = title;

this.picture = picture;

this.district = district;

this.tag = tag;

this.cost = cost;

var firstSpace = 290;

for (let i = 290; i > 250; i--) {

if(info[i] == ' ') {

firstSpace = i;

break;

}

}

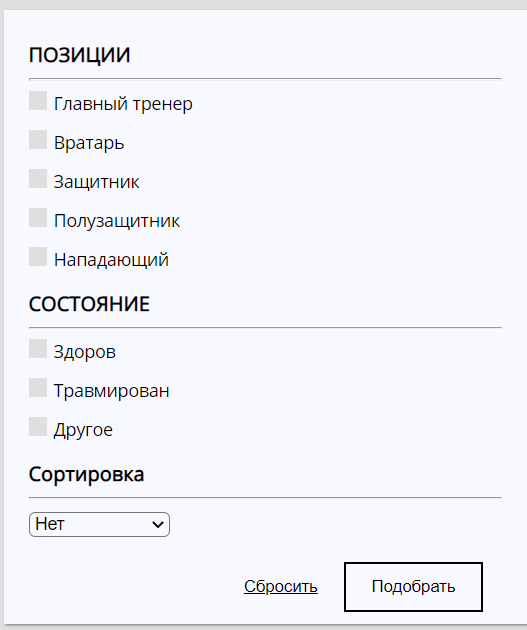
this.info = info.slice(0, firstSpace) + "...";

}

Объект имеет следующие поля:

* title – фамилия игрока.
* picture –изображения игрока.
* district – позиция, на которой играет игрок.
* tag – состояние игрока.
* cost – стоимость игрока.
* info – краткая информация об игроке.

Однако, на страницу выводятся только фамилия, изображение и краткая информация при помощи функции ToBlock(), принимающей параметром объект «игрок». Остальные поля используются для последующей фильтрации массива объектов при помощи меню фильтров (рисунок 8).



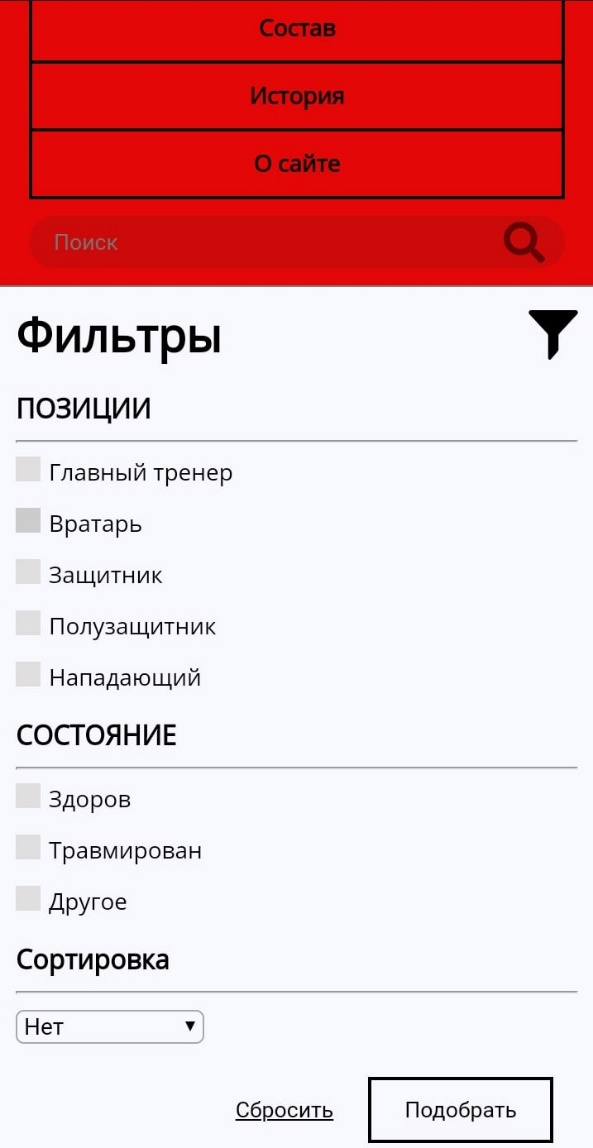


Рисунок 8 – фильтры (сверху – версия сайта для пк, снизу –   
мобильная версия)

В мобильной версии фильтры по умолчанию спрятаны, появляются при нажатии на надпись «Фильтры», которая расположена под «шапкой» страницы.

Если переход на страницу «Состав» осуществился через выбор позиции игрока на поле в выпадающем меню (рисунок 2), то происходит следующее. При нажатии на позицию вызывается функция send(), отправляющая в локальное хранилище переменную-индекс checkbox’а позиции в массиве при помощи метода localStorage.setItem(). В свою очередь, на странице состав после загрузки страницы вызывается функция check(), получающая из локального хранилища индекс позиции через метод getItem() и устанавливающая соответствующей позиции checkbox в состояние "выбрано".

Далее пользователь может задать критерии фильтрации и сортировки. Для того, чтобы они выполнились, необходимо нажать кнопку «Подобрать». При нажатии вызывается функция Filters(). Алгоритм ее работы следующий. Имеется две группы критериев фильтрации: позиция на поле и состояние здоровья. А также два варианта сортировки: по стоимости и по алфавиту. Массив игроков фильтруется по каждой группе критериев последовательно. При этом, если игрок соответствует хотя бы одному из выбранных критериев одной группы, то он добавляется в отфильтрованный массив. После фильтрации производится сортировка игроков по заданному параметру и при помощи функции ToBlock() конечный массив выводится на экран. В случае, если массив пуст, выводится надпись «Ничего не найдено!» (рисунок 9).

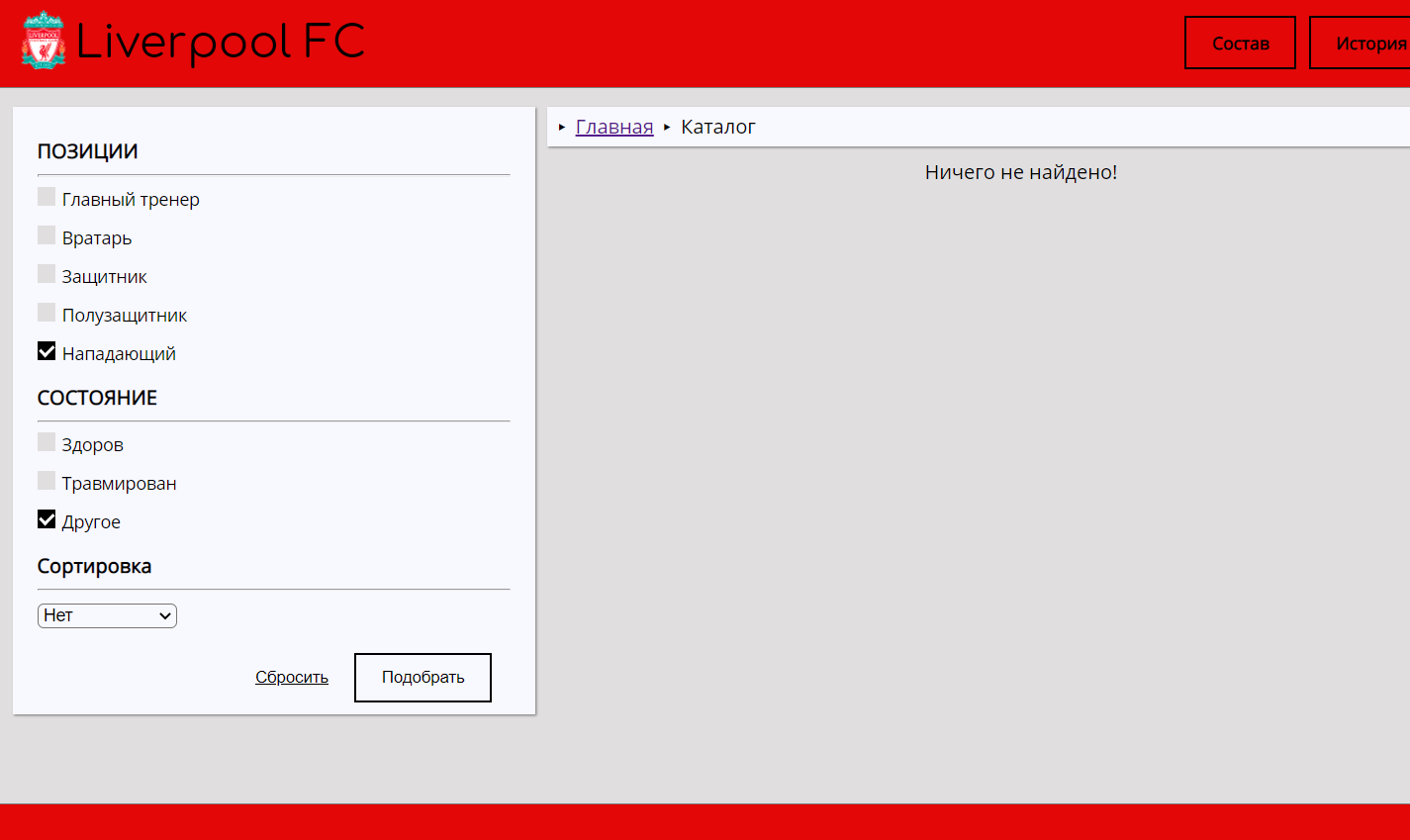


Рисунок 9 – в составе не найдено игроков, соответствующих критериям фильтрации

Если же переход на страницу «Состав» осуществился при вводе поискового запроса в поисковую форму, то поисковый запрос записывается в локальное хранилище, и при создании первоначального массива происходит фильтрация по полученному из локального хранилища запросу. Игрок попадает в результирующий массив, если его поле title содержит подстроку, соответствующую поисковому запросу. Далее при помощи фильтров мы можем работать с массивом игроков, найденных по запросу, пока не нажмем кнопку «сбросить», которая сбрасывает фильтры и возвращает на экран первоначальный массив игроков.

Также на странице «Состав» присутствует специальная кнопка-якорь «наверх», позволяющая перейти к началу каталога (рисунок 10).



Рисунок 10 – кнопка «Наверх»

### **Страница «история»**

На данной странице расположены статьи из истории Ливерпуля, подкрепляющиеся наглядными иллюстрациями о выбранном периоде клуба (рисунок 11).

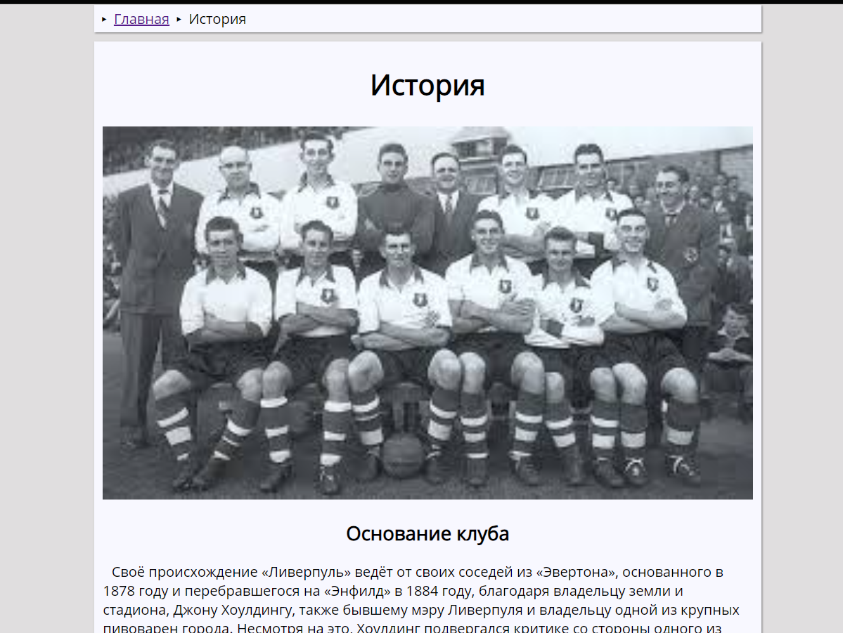


Рисунок 11 – страница «История»

### **Страница «О сайте»**

На данной странице расположена текстовая информация о цели создания сайта (рисунок 12).



Рисунок 12 – «О сайте»

### **Страница «Игрок»**

Страница «игрок» представляет собой структуру для отображения информации о игроках на ее основе (Рисунок 13).



Рисунок 13 – страница «Игрок»

Переход на эту страницу возможен при нажатии на фамилию игрока в каталоге. В этом случае в локальное хранилище передается фамилия игрока. В скрипте player.js, который подключен к странице «Игрок», происходит парсинг players.xml в DOM (процесс такой же, как и в каталоге «Состав», только без преобразования к объекту). Далее, в xml DOM выполняется поиск игрока по фамилии, и, когда он найден, на страницу выводится вся информация об экземпляре.

# **Заключение**

В ходе выполнения данной курсовой работы были изучены некоторые особенности JavaScript, разработаны состав и структура веб-сайта «Фан сайт футбольного клуба Liverpool», веб-страницы сайта были наполнены тематическим контентом, также был создан удобный пользовательский интерфейс.

Мною были изучены особенности таких технологий, как локальное хранилище и AJAX-запросы.

Локальное хранилище браузера активно использовалось при разработке для передачи примитивных данных между веб-страницами. Хотя оно и не подходит для хранения конфиденциальных данных, в данном случае это не является проблемой, так как оно использовалось для хранения индекса checkbox’а и строки поискового запроса. В ходе разработки веб-сайта выяснилось, что браузеры Edge и Internet Explorer имеют некоторые проблемы, связанные с локальным хранилищем.

AJAX-запросы использовались при получении с сервера xml-файлов, таких как student.xml и players.xml для последующего парсинга на веб-страницы сайта.

Особенностью данного сайта является то, что нам не нужно вручную добавлять и удалять страницы для игроков: мы можем добавить или удалить его в xml-документе, наполнить его информацией о игроке, и игрок появится на сайте как в каталоге, так и в виде отдельной странице.

Данный веб-сайт был размещен на бесплатном хостинге GitHub Pages.

# **Список источников**

1. Современный учебник JavaScript [Электронный ресурс] Режим доступа: <https://learn.javascript.ru/> - Дата доступа: 05.12.2020.
2. Справочник HTML [Электронный ресурс] Режим доступа: <http://htmlbook.ru/> - Дата доступа: 08.12.2020.
3. Веб-сайт современной документации Mozilla Developer Network [Электронный ресурс] Режим доступа: <https://developer.mozilla.org/> - Дата доступа: 08.12.2020
4. Жиляк Н.А. Информационные технологии. Разработка Web-сайта на основе HTML с использованием JavaScript : методические указания к выполнению курсовых / [сост. Н. А. Жиляк]. - Минск : БГТУ, 2012. - 59, [1] c.
5. Интернет-энциклопедия Wikipedia [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/ - Дата доступа: 05.12.2020.

# **Приложение**

## **Приложение А**

Структура веб-страницы на примере index.html:

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="utf-8">

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

<title>Фан-Страница Liverpool</title>

<link rel="shortcut icon" href="images/lv.jpg" type="image/x-icon">

<!--Подключаемые стили-->

<link rel="stylesheet" href="styles/main.css">

<link rel="stylesheet" href="styles/slick.css">

<link rel="stylesheet" href="styles/slick-theme.css">

<!--Подключаемые скрипты-->

<script src="scripts/jquery-3.4.1.min.js"></script>

<script src="scripts/slick.min.js"></script>

<script src="scripts/counter.js"></script>

<script src="scripts/check-district.js"></script>

<script src="scripts/search.js"></script>

<!--Скрипт галереи-карусели-->

<script>

$(document).ready(function(){

$('.gallery').slick({

slidesToScroll: 1,

autoplay: true,

autoplaySpeed: 3000,

speed: 750,

});

});

</script>

</head>

<body>

<div class="wrapper">

<!--Шапка страницы-->

<div class="header">

<a href="index.html">

<div class="logo"><img src="images/lvp.png" alt="Логотип сайта"></div>

<div class="title">Liverpool FC</div>

</a>

<div class="menu">

<div class="button" id="catalog">

<a href="catalog.html" onclick="send(null)">Состав</a>

<!--Выпадающее меню позиций каталога. При нажатии на соответствующую гиперссылку вызывается функция send(), принимающая аргументом индекс позиции в форме фильтров и записывающая его в локальное хранилище для последующего получения при помощи функции check() в каталоге-->

<ul class="districts">

<li><a href="catalog.html" onclick="send(0)">Главный тренер</a></li>

<li><a href="catalog.html" onclick="send(1)">Вратари</a></li>

<li><a href="catalog.html" onclick="send(2)">Защитники</a></li>

<li><a href="catalog.html" onclick="send(3)">Полузащитники</a></li>

<li><a href="catalog.html" onclick="send(4)">Нападающие</a></li>

</ul>

</div>

<div class="button"><a href="history.html">История</a></div>

<div class="button"><a href="about.html">О сайте</a></div>

</div>

<!--Форма для поиска-->

<form class="search" action="catalog.html">

<input type="search" name="input1" placeholder="Поиск" class="input" />

<button class="submit" onclick="Search()"></button>

</form>

</div>

<!--Основной блок-->

<div class="main">

<div class="content">

<!--Хлебные крошки-->

<ul class="bread\_crumbs">

<li>&rtrif; Главная </li>

</ul>

<!--Основная информация на странице-->

<div class="article">

<h2 class="heading">Главная страница</h2>

<!--Галерея-->

<div class="gallery">

<div class="image">

<a href="history.html"> <img src="images/history.jpg" title="История клуба"/>

</a>

</div>

<div class="image">

<a href="catalog.html"> <img src="images/line\_up.jpg" title="Состав" />

</a>

</div>

<div class="image">

<a href="https://www.nike.com/ru/"> <img src="images/form.jpeg" title="Футбольная атрибутика"/>

</a>

</div>

</div>

<p>«Ливерпуль» (полное название — Футбольный клуб «Ливерпуль», англ. Liverpool City Football Club) — английский профессиональный футбольный клуб из одноимённого города, расположенного в графстве Мерсисайд. Был основан в 1892 году; в следующем году вступил в Футбольную лигу Англии. С момента своего создания проводит домашние игры на стадионе «Энфилд». Стадион был построен в 1884 году, на нём изначально играл другой ливерпульский футбольный клуб, «Эвертон». В настоящий момент «Энфилд» вмещает порядка пятидесяти четырёх тысяч болельщиков.</p>

<p>«Ливерпуль» является 19-кратным чемпионом Англии, 7-кратным обладателем Кубка Англии, 8-кратным обладателем Кубка лиги, 15-кратным обладателем Суперкубка Англии, 3-кратным обладателем Кубка УЕФА, 6-кратным победителем Лиги чемпионов, 4-кратным обладателем Суперкубка УЕФА. В общей сложности «Ливерпуль» выигрывал еврокубки тринадцать раз, что является четвёртым результатом после мадридского «Реала», «Милана» и «Барселоны». Самый успешный период в истории клуба приходится на 1970-е и 1980-е годы, когда командой руководили Билл Шенкли и Боб Пейсли, которые совместно на двоих выиграли 11 чемпионских титулов и 7 европейских трофеев.</p>

<p>Принципиальными соперниками с давних лет являются английские команды «Эвертон» и «Манчестер Юнайтед», с которыми разыгрываются мерсисайдское и северо-западное дерби соответственно. Гимном клуба является песня You’ll Never Walk Alone, перепетая в дальнейшем ливерпульской группой Gerry & The Pacemakers.</p>

<p>По данным Forbes на 2018 год, «Ливерпуль» занимал восьмое место в списке самых дорогих футбольных клубов мира и оценивался в сумму 1,944 млрд долларов США. По данным аудиторской компании Deloitte, по итогам сезона 2016/17 «Ливерпуль» занимал 9-ю строчку в рейтинге футбольных клубов с ежегодным доходом в размере 424,2 млн евро.</p>

<p>На сайте вы можете найти актуальную информацию о составе, историю гранда мирового футбола, а так же расположение магазинов футбольной атрибутики.</p>

</div>

</div>

</div>

<!--Подвал-->

<div class="footer">

<!--Счетчик Яндекс.Метрика-->

<div class="counter">

<!-- Yandex.Metrika informer -->

<a href="https://metrika.yandex.ru/stat/?id=70486777&amp;from=informer"

target="\_blank" rel="nofollow"><img src="https://informer.yandex.ru/informer/70486777/3\_1\_FFFFFFFF\_EFEFEFFF\_0\_pageviews"style="width:88px; height:31px; border:0;" alt="Яндекс.Метрика" title="Яндекс.Метрика: данные за сегодня (просмотры, визиты и уникальные посетители)" class="ym-advanced-informer" data-cid="70486777" data-lang="ru" /></a>

<!-- /Yandex.Metrika informer -->

<!-- Yandex.Metrika counter -->

<script type="text/javascript" > (function(m,e,t,r,i,k,a){m[i]=m[i]||function(){(m[i].a=m[i].a||[]).push(arguments);m[i].l=1\*new Date();k=e.createElement(t),a=e.getElementsByTagName(t)[0],k.async=1,k.src=r,a.parentNode.insertBefore(k,a)})

(window, document, "script", "https://mc.yandex.ru/metrika/tag.js", "ym");

ym(70486777, "init", {

clickmap:true,

trackLinks:true,

accurateTrackBounce:true

});

</script>

<noscript><div><img src="https://mc.yandex.ru/watch/70486777" style="position:absolute; left:-9999px;" alt="" /></div></noscript>

<!-- /Yandex.Metrika counter --><!--Таблица информации о студенте-->

<table id="author">

<tr>

<td>ФИО</td>

<td>Факультет</td>

<td>Специальность</td>

<td>Группа</td>

<td>Год</td>

</tr>

<tr>

<td></td>

<td></td>

<td></td>

<td></td>

<td></td>

</tr>

</table>

<!--Скрипт-парсер информации о студенте из файла student.xml-->

<script>

var xmlhttp;

if (window.XMLHttpRequest)

{ // для IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

}

else

{ // для IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.open("GET","https://raw.githubusercontent.com/SNikita24/Coursework/main/student.xml",false);

xmlhttp.overrideMimeType('application/xml');

xmlhttp.send();

var xmlDoc = xmlhttp.responseText;

var parser = new DOMParser();

var doc = parser.parseFromString(xmlDoc, "application/xml");

var string = document.getElementById("author").getElementsByTagName("tr")[1].getElementsByTagName("td");

for (let i = 0; i < string.length; i++) {

string[i].innerHTML = doc.getElementsByTagName("student")[0].childNodes[2 \* i + 1].textContent;

}

</script>

</div>

</div>

</body>

</html>

## **Приложение Б**

Javascript-файл catalog.js

'use strict';

//Класс игрок

function Player(title, picture, info, district, tag, cost) {

this.title = title;

this.picture = picture;

this.district = district;

this.tag = tag;

this.cost = cost;

var firstSpace = 290;

for (let i = 290; i > 250; i--) {

if(info[i] == ' ') {

firstSpace = i;

break;

}

}

this.info = info.slice(0, firstSpace) + "...";

}

//Функция, получающая из локального хранилища индекс позиции и устанавливающая соответствующий позиции checkbox в состояние "выбрано"

function check() {

var districts = document.getElementById("filters");

var number = localStorage.getItem("district");

if (number >= 0 && number < districts.location.length) {

districts.location[number].checked = true;

}

}

//Парсер игроков из Xml через XMLHttpRequest

function PlayerParser() {

var xmlhttp;

if (window.XMLHttpRequest)

{ // для IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

}

else

{ // для IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.open("GET","https://raw.githubusercontent.com/SNikita24/Coursework/main/players.xml",false);

xmlhttp.overrideMimeType('application/xml');

xmlhttp.send();

var xmlDoc = xmlhttp.responseText;

var parser = new DOMParser();

return parser.parseFromString(xmlDoc, "application/xml");

}

//Функция вывода экземпляра класса игрок на страницу

function ToBlock(player) {

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

main.setAttribute("class", "player");

var refTitle = document.createElement("a");

refTitle.innerHTML = player.title;

refTitle.setAttribute("class", "name");

refTitle.setAttribute("href", "player.html");

refTitle.setAttribute("target", "\_blank");

//При нажатии на ссылку записываем ее текст в локальное хранилище

refTitle.onclick = function(Event) {

localStorage.setItem("title", Event.currentTarget.innerHTML);

}

main.appendChild(refTitle);

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

divInfo.setAttribute("class", "info");

var hr = document.createElement("hr");

var img = document.createElement("img");

img.setAttribute("src", "images/" + player.picture);

img.setAttribute("alt", player.picture);

img.setAttribute("title", player.title);

var span = document.createElement("span");

span.setAttribute("class", "about");

span.innerHTML = player.info;

divInfo.appendChild(hr);

divInfo.appendChild(img);

divInfo.appendChild(span);

main.appendChild(refTitle);

main.appendChild(divInfo);

document.getElementById("player-container").appendChild(main);

}

//Сброс формы и поискового запроса

function Reset() {

localStorage.removeItem("whatToSearch");

if (searchedPlayersByName != null) {

searchedPlayersByName.length = 0;

}

for (let i = 0; i < document.getElementsByName("location").length; i++) {

document.getElementsByName("location")[i].checked = false;

}

for (let i = 0; i < document.getElementsByName("subject").length; i++) {

document.getElementsByName("subject")[i].checked = false;

}

document.getElementsByName("cost")[0].checked = false;

Filters();

}

//Фильтрация массива игроков по параметрам из формы

function Filters() {

var searchedPlayers = new Array();

var namesOfDistricts = new Array();

var districts = document.getElementsByName("location");

for (let i = 0, j = 0; i < districts.length; i++) {

if (districts[i].checked) {

namesOfDistricts[j] = districts[i].value;

j++;

}

}

var namesOfSubjects = new Array();

var subjects = document.getElementsByName("subject");

for (let i = 0, j = 0; i < subjects.length; i++) {

if (subjects[i].checked) {

namesOfSubjects[j] = subjects[i].value;

j++;

}

}

var IsFree = document.getElementsByName("cost")[0].checked;

//Если есть непустой массив, отфильтрованный по поиску, работаем с ним

if (searchedPlayersByName != null && searchedPlayersByName.length != 0) {

var searchedPlayers = searchedPlayersByName.slice();

}

else {

var searchedPlayers = players.slice();

}

//Если элемент соответствует хотя бы одному параметру, добавляем в отфильтрованный массив

if (namesOfDistricts.length != 0) {

searchedPlayers = searchedPlayers.filter(function(item) {

for (let i = 0; i < namesOfDistricts.length; i++) {

if (item.district == namesOfDistricts[i]) {

return true;

}

}

return false;

});

}

if(namesOfSubjects.length != 0){

searchedPlayers = searchedPlayers.filter(function(item) {

for (let i = 0; i < namesOfSubjects.length; i++) {

if (item.tag == namesOfSubjects[i]) {

return true;

}

}

return false;

});

}

//Сортировка игроков по заданному в select параметру

switch (document.getElementsByTagName("select")[0].selectedIndex) {

case 0: break;

case 1: searchedPlayers.sort(function(a, b) {

if (a.title < b.title) {

return -1;

}

if (a.title > b.title) {

return 1;

}

return 0;

});

break;

case 2: searchedPlayers.sort(function(a, b) {

return a.cost - b.cost;

});

break;

default: break;

}

document.getElementById("player-container").innerHTML = "";

if (searchedPlayers.length == 0) {

document.getElementById("player-container").innerHTML = "Ничего не найдено!";

}

else {

for (let i = 0; i < searchedPlayers.length; i++) {

ToBlock(searchedPlayers[i]);

}

}

}

var doc = PlayerParser();

var players = [];

//Инициализация массива экземпляров игроков, получая поля для конструктора из XmlDOM

for (let i = 0; i < doc.getElementsByTagName("player").length; i++) {

players[i] = new Player(doc.getElementsByTagName("player")[i].getElementsByTagName("title")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("picture")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("info")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("district")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("subjects")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("cost")[0].textContent

}

var searchRequest = localStorage.getItem("whatToSearch");

//Если передан поисковый запрос, ищем вхождение запроса в игроков

if(searchRequest != null && searchRequest.length != 0){

var searchedPlayersByName = players.filter(function(item){

var name = item.title.toLowerCase();

var search = searchRequest.toLowerCase();

//Объявление для IE (не поддерживает метод includes())

if (!String.prototype.includes) {

String.prototype.includes = function(search, start) {

if (typeof start !== 'number') {

start = 0;

}

if (start + search.length > this.length) {

return false;

}

else {

return this.indexOf(search, start) !== -1;

}

};

}

if(name.includes(search)){

return true;

}

return false;

});

localStorage.removeItem("whatToSearch");

if(searchedPlayersByName.length == 0){

document.getElementById("player-container").innerHTML = "Ничего не найдено!";

}

else{

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

ToBlock(searchedPlayersByName[i]);

}

}

}

else {

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

ToBlock(players[i]);

}

} 'use strict';

//Класс игрок

function Player(title, picture, info, district, tag, cost) {

this.title = title;

this.picture = picture;

this.district = district;

this.tag = tag;

this.cost = cost;

var firstSpace = 290;

for (let i = 290; i > 250; i--) {

if(info[i] == ' ') {

firstSpace = i;

break;

}

}

this.info = info.slice(0, firstSpace) + "...";

}

//Функция, получающая из локального хранилища индекс позиции и устанавливающая соответствующий позиции checkbox в состояние "выбрано"

function check() {

var districts = document.getElementById("filters");

var number = localStorage.getItem("district");

if (number >= 0 && number < districts.location.length) {

districts.location[number].checked = true;

}

}

//Парсер игроков из Xml через XMLHttpRequest

function PlayerParser() {

var xmlhttp;

if (window.XMLHttpRequest)

{ // для IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

}

else

{ // для IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.open("GET","https://raw.githubusercontent.com/SNikita24/Coursework/main/players.xml",false);

xmlhttp.overrideMimeType('application/xml');

xmlhttp.send();

var xmlDoc = xmlhttp.responseText;

var parser = new DOMParser();

return parser.parseFromString(xmlDoc, "application/xml");

}

//Функция вывода экземпляра класса игрок на страницу

function ToBlock(player) {

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

main.setAttribute("class", "player");

var refTitle = document.createElement("a");

refTitle.innerHTML = player.title;

refTitle.setAttribute("class", "name");

refTitle.setAttribute("href", "player.html");

refTitle.setAttribute("target", "\_blank");

//При нажатии на ссылку записываем ее текст в локальное хранилище

refTitle.onclick = function(Event) {

localStorage.setItem("title", Event.currentTarget.innerHTML);

}

main.appendChild(refTitle);

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

divInfo.setAttribute("class", "info");

var hr = document.createElement("hr");

var img = document.createElement("img");

img.setAttribute("src", "images/" + player.picture);

img.setAttribute("alt", player.picture);

img.setAttribute("title", player.title);

var span = document.createElement("span");

span.setAttribute("class", "about");

span.innerHTML = player.info;

divInfo.appendChild(hr);

divInfo.appendChild(img);

divInfo.appendChild(span);

main.appendChild(refTitle);

main.appendChild(divInfo);

document.getElementById("player-container").appendChild(main);

}

//Сброс формы и поискового запроса

function Reset() {

localStorage.removeItem("whatToSearch");

if (searchedPlayersByName != null) {

searchedPlayersByName.length = 0;

}

for (let i = 0; i < document.getElementsByName("location").length; i++) {

document.getElementsByName("location")[i].checked = false;

}

for (let i = 0; i < document.getElementsByName("subject").length; i++) {

document.getElementsByName("subject")[i].checked = false;

}

document.getElementsByName("cost")[0].checked = false;

Filters();

}

//Фильтрация массива игроков по параметрам из формы

function Filters() {

var searchedPlayers = new Array();

var namesOfDistricts = new Array();

var districts = document.getElementsByName("location");

for (let i = 0, j = 0; i < districts.length; i++) {

if (districts[i].checked) {

namesOfDistricts[j] = districts[i].value;

j++;

}

}

var namesOfSubjects = new Array();

var subjects = document.getElementsByName("subject");

for (let i = 0, j = 0; i < subjects.length; i++) {

if (subjects[i].checked) {

namesOfSubjects[j] = subjects[i].value;

j++;

}

}

var IsFree = document.getElementsByName("cost")[0].checked;

//Если есть непустой массив, отфильтрованный по поиску, работаем с ним

if (searchedPlayersByName != null && searchedPlayersByName.length != 0) {

var searchedPlayers = searchedPlayersByName.slice();

}

else {

var searchedPlayers = players.slice();

}

//Если элемент соответствует хотя бы одному параметру, добавляем в отфильтрованный массив

if (namesOfDistricts.length != 0) {

searchedPlayers = searchedPlayers.filter(function(item) {

for (let i = 0; i < namesOfDistricts.length; i++) {

if (item.district == namesOfDistricts[i]) {

return true;

}

}

return false;

});

}

if(namesOfSubjects.length != 0){

searchedPlayers = searchedPlayers.filter(function(item) {

for (let i = 0; i < namesOfSubjects.length; i++) {

if (item.tag == namesOfSubjects[i]) {

return true;

}

}

return false;

});

}

//Сортировка игроков по заданному в select параметру

switch (document.getElementsByTagName("select")[0].selectedIndex) {

case 0: break;

case 1: searchedPlayers.sort(function(a, b) {

if (a.title < b.title) {

return -1;

}

if (a.title > b.title) {

return 1;

}

return 0;

});

break,

case 2: searchedPlayers.sort(function(a, b) {

return a.cost - b.cost;

});

break;

default: break;

}

document.getElementById("player-container").innerHTML = "";

if (searchedPlayers.length == 0) {

document.getElementById("player-container").innerHTML = "Ничего не найдено!";

}

else {

for (let i = 0; i < searchedPlayers.length; i++) {

ToBlock(searchedPlayers[i]);

}

}

}

var doc = PlayerParser();

var players = [];

//Инициализация массива экземпляров игроков, получая поля для конструктора из XmlDOM

for (let i = 0; i < doc.getElementsByTagName("player").length; i++) {

players[i] = new Player(doc.getElementsByTagName("player")[i].getElementsByTagName("title")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("picture")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("info")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("district")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("subjects")[0].textContent,

doc.getElementsByTagName("player")[i].getElementsByTagName("cost")[0].textContent

);

}

var searchRequest = localStorage.getItem("whatToSearch");

//Если передан поисковый запрос, ищем вхождение запроса в игроков

if(searchRequest != null && searchRequest.length != 0){

var searchedPlayersByName = players.filter(function(item){

var name = item.title.toLowerCase();

var search = searchRequest.toLowerCase();

//Объявление для IE (не поддерживает метод includes())

if (!String.prototype.includes) {

String.prototype.includes = function(search, start) {

if (typeof start !== 'number') {

start = 0;

}

if (start + search.length > this.length) {

return false;

}

else {

return this.indexOf(search, start) !== -1;

}

};

}

if(name.includes(search)){

return true;

}

return false;

});

localStorage.removeItem("whatToSearch");

if(searchedPlayersByName.length == 0){

document.getElementById("player-container").innerHTML = "Ничего не найдено!";

}

else{

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

ToBlock(searchedPlayersByName[i]);

}

}

}

else {

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

ToBlock(players[i]);

}

}

## **Приложение В**

Document Type Definition и пример структуры игрока в файле players.xml <?xml version="1.0" encoding="UTF-8"?>

<!--Xml-документ, хранящий информацию о игроках, размещенных на сайте-->

<!DOCTYPE players [

<!ELEMENT players (player\*)>

<!ELEMENT player (title, info, picture+, district, subjects, time, home, email, cost, number)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT info (#PCDATA)>

<!ELEMENT picture (#PCDATA)>

<!ELEMENT district (#PCDATA)>

<!ELEMENT subjects (#PCDATA)>

<!ELEMENT time (#PCDATA)>

<!ELEMENT home (#PCDATA)>

<!ELEMENT email (#PCDATA)>

<!ELEMENT cost (#PCDATA)>

<!ELEMENT number (#PCDATA)>

]>

<players>

<player>

<title>Алисон</title>

<info>

Алиссон Рамзес Бекер (порт. Alisson Ramses Becker; родился 2 октября 1992 года в Нову-Амбургу, Бразилия), более известный как Алиссон — бразильский футболист немецкого происхождения, вратарь клуба «Ливерпуль» и сборной Бразилии. Участник чемпионата мира 2018 года.

Алиссон — воспитанник клуба «Интернасьонал». 17 февраля 2013 года в матче Лиги Гаушу против «Крузейро» из Порту-Алегри он дебютировал за основную команду. 25 августа в поединке против «Гояса» Алиссон дебютировал в бразильской Серии A. В 2015 году после того, как Дида завершил карьеру, он стал основным вратарём команды. В составе «Интернасьонала» Алиссон трижды выиграл Лигу Гаушу.

Уже в начале 2018 года заинтересованность во вратаре проявили испанская «Барселона», французский «Пари Сен-Жермен» и английский «Ливерпуль». Летом того же года Алиссон подписал шестилетний контракт с последним. Сумма трансфера составила 66,8 млн фунтов (72,5 млн евро), что сделало Бекера самым дорогим вратарём в истории (прежнее достижение принадлежало итальянцу Джанлуиджи Буффону и продержалось 17 лет).

</info>

<picture>alison.jpg</picture>

<district>Вратарь</district>

<subjects>Травмирован</subjects>

<time>28 </time>

<home>Бразилия</home>

<email>https://www.instagram.com/alissonbecker/</email>

<cost>80.0</cost>

<number>1</number>

</player>

</players>