**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ ЦЕНТР ПРОФЕССИОНАЛЬНЫХ КВАЛИФИКАЦИЙ И СОДЕЙСТВИЯ ТРУДОУСТРОЙСТВУ «ПРОФЕССИОНАЛ»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к итоговой аттестационной работе на тему

**«Разработка web-ресурса с использованием технологий HTML, CSS, JavaScript»**

слушателя Ковнира Дмитрия Андреевича группы №: 0155

программы профессиональной переподготовки

«Frontend разработка»

Москва, 2023

Оглавление

[Постановка задачи и план работы. 3](#_Toc103884852)

[ОСНОВНАЯ ЧАСТЬ. 4](#_Toc103884853)

[Назначение веб-ресурса. 4](#_Toc103884854)

[Описание этапов разработки, описание функционала с приложением листингов исходного программного кода основной функции, структурных модулей, пример кода-разметки **4**](#_Toc103884855)

[Список литературы 11](#_Toc103884856)

# ПОСТАНОВКА ЗАДАЧИ И ПЛАН РАБОТ.

Целью проекта является разработка веб-портала, на котором будут представлены игры "Крестики-нолики" и "Угадай число". Веб-портал будет предоставлять пользователям возможность играть онлайн в любую из представленных игр на выбор без необходимости установки дополнительного программного обеспечения.

План работы включает в себя следующие этапы:

1. Проектирование веб-портала и его функционала.
2. Разработка игровой логики для каждой из игр с использованием JavaScript.
3. Создание HTML и CSS для представления игр и интерфейса веб-портала.
4. Тестирование веб-портала и его функционала.
5. Публикация веб-портала.

# ОСНОВНАЯ ЧАСТЬ.

# Назначение веб-ресурса.

Веб-ресурс представляет собой игровой портал, который предлагает пользователям возможность играть в различные игры прямо в браузере. Пользователи могут выбрать на портале игру и начать играть. В настоящее время на портале доступны следующие игры: "Крестики-нолики" и "Угадай число". Игра "Космическая защита" находится в разработке. Игры можно добавлять по мере их выпуска.

## Описание этапов разработки, описание функционала с применением листингов исходного программного кода основной функции, структурных модулей, пример кода-разметки

**Этапы разработки.**

* Были определены требования к веб-порталу и его функционалу, включая интерфейс выбора игры и игровую логику каждой из игр.
* Для каждой из игр была разработана своя игровая логика с использованием JavaScript.
* Для создания внешнего вида портала и представления игр использовались HTML и CSS. Был создан центральный список игр со ссылками на каждую из игр, а так же реализован поиск по играм.
* Проведена проверка правильности работы веб-портала, его функционала и каждой из игр, исправлены ошибки и недочеты.
* После тестирования и исправления всех ошибок веб-портал был опубликован и стал доступным для пользователей, а так же выложен на GitHub посредством системы контроля версий Git по адресу \_\_\_\_\_\_\_. Тестирование в автоматическом режиме не производилось.

В результате был разработан веб-портал, на котором были представлены различные игры, каждая из которых имеет свою игровую логику и предоставляет пользователю интерактивный игровой процесс.

**Описание реализации веб-портала**

Пример HTML-кода, который создает список игр на главной странице веб-портала:

<div class="search-bar">

    <input type="text" class="search-input" placeholder="Введите название игры">

  </div>

  <ul class="game-list">

    <li class="game-list-item" href="cross.html">

      <a href="newcross.html">Крестики-Нолики</a>

    </li>

    <li class="game-list-item">

      <a href="guess.html">Угадай число</a>

    </li>

    <li class="game-list-item">

      <a href="shipdefence.html">Космическая защита<br><sub text="center">(в разработке)</sub></a>

    </li>

  </ul>

В этом примере используется структура списка <ul>, каждый элемент которого <li> представляет собой игру. Каждая игра включает в себя ссылку <a>, ведущую к HTML-странице этой игры.

Следующий пример JavaScript кода реализует функцию поиска по играм:

  <script>

    let searchInput = document.querySelector('.search-input');

    let gameItems = document.querySelectorAll('.game-list-item');

    searchInput.addEventListener('input', function (event) {

      let wordToSearch = event.target.value.toLowerCase();

      gameItems.forEach(function (item) {

        let gameName = item.querySelector('a').textContent.toLowerCase();

        if (gameName.includes(wordToSearch)) {

          item.style.display = 'flex';

        } else {

          item.style.display = 'none';

        }

      });

    });

  </script>

Этот код использует DOM API браузера для обработки пользовательского ввода в поле поиска и отображения только тех игр, которые соответствуют введенному тексту. В первых двух строках с помощью метода querySelector и querySelectorAll получаются элементы поля ввода и элементы списка игр соответственно. Затем добавляется обработчик события input, который вызывается каждый раз, когда пользователь вводит текст в поле поиска. В обработчике события сначала получается текст, введенный пользователем, и преобразуется в нижний регистр. Затем для каждого элемента списка игр выполняется проверка посредством метода includes: если имя игры включает в себя введенный текст, то элемент списка становится видимым, в противном случае - скрытым.

**Описание страниц игр и примеры логики JavaScript**

**Крестики-нолики**

Реализация игры представляет собой интерактивную страницу для игры в "Крестики-нолики". Страница позволяет пользователям сыграть в классическую настольную игру прямо в браузере, без необходимости загружать и устанавливать специальные приложения. Пользователи могут играть друг с другом, делая ходы по очереди.

Функционал игры реализован посредством HTML-файла (newcross.html), CSS-файла (newcross.css) и JavaScript файла (newcross.js).

HTML-файл (**newcross.html)** представляет собой интерфейс для игры. Пример кода страницы:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" type="text/css" href="newcross.css">

  <title>Крестики-Нолики</title>

</head>

<body>

  <div id="board"></div>

  <button id="reset">Сбросить</button>

  <script src="newcross.js"></script>

</body>

</html>

Код страницы состоит из следующих основных элементов:

**Игровая доска**:

<div id="board"></div>

Это основная часть интерфейса, где происходит игра. Этот <div> будет заполнен игровыми ячейками динамически с помощью JavaScript. Каждая ячейка будет отдельным <div> элементом, и при нажатии на него, он будет заполняться символом текущего игрока (крестиком или ноликом).

**Кнопка сброса:**

<button id="reset">Сбросить</button>

Эта кнопка позволяет пользователю сбросить игровую доску и начать игру заново. При нажатии на эту кнопку все ячейки игровой доски очищаются, и игра начинается сначала.

**Подключенные CSS и JS файлы:**

В разделе <head> подключается файл стилей (newcross.css), который определяет внешний вид игровой доски и других элементов интерфейса. В разделе <body> подключается JavaScript файл (newcross.js), который содержит логику игры.

JavaScript файл управляет поведением игры, включая создание игровых ячеек, обработку ходов игроков, определение победителя и обработку события нажатия на кнопку сброса. CSS-файл (newcross.css) определяет стили для игровой доски и кнопки сброса.

JavaScript файл (newcross.js), представляет собой описание логики игры. **Описание логики с примерами кода JavaScript**

Работа игры начинается с рисования игровой доски посредством вызова функции resetBoard(). Данная функция так же используется для сброса игровой доски, чтобы начать игру заново.

function resetBoard() {

    board = Array(9).fill(null);

    player = 'X';

    drawPlayBoard();

}

После того как массив board, представляющий из себя игровое поле, очищен и присвоено текущее значение первому игроку, производится рисование игровой доски посредством функции drawPlayBoard(). Данная функция рисует не только начальное состояние игровой доски, но также используется в дальнейшем с учётом сделанных ходов.

function drawPlayBoard() {

    let boardDiv = document.getElementById('board');

    boardDiv.innerHTML = '';

    board.forEach((value, index) => {

        boardDiv.innerHTML += `<div onclick="makeMove(${index})">${value ? value : ''}</div>`;

    });

}

После того как игрок нажимает на игровое поле, вызывается функция makeMove() с соответствующим индексом в аргументе. В функции производится проверка на возможность хода в данную ячейку, присвоение соответствующего значения хода, проверка на победителя посредством функции checkWinner(), проверка на “Ничью” и смена игрока. А так же рисуется доска с обновлёнными данными.

function makeMove(index) {

    if (!board[index]) {

        board[index] = player;

        if (checkWinner()) {

            alert(`Выиграл игрок ${player}`);

            resetBoard();

        } else {

            if (board.every(cell => cell)) {

                alert('Ничья!');

                resetBoard();

            } else {

                player = player === 'X' ? 'O' : 'X';

            }

        }

        drawPlayBoard();

    }

}

Функция checkWinner() проверяет после хода игрока наличие победителя. Для этого используется двумерный массив. Массив состоит из восьми элементов, каждый из которых является массивом из индексов массива board составляющих выигрышную комбинацию. Функция возвращает true или false при успешном выполнении функций с использованием методов some и every. Иными словами возвращает true если хотя бы один из “подмассивов” содержит все элементы равные значению текущего игрока.

function checkWinner() {

    let winningCombinations = [

        [0, 1, 2],

        [3, 4, 5],

        [6, 7, 8],

        [0, 3, 6],

        [1, 4, 7],

        [2, 5, 8],

        [0, 4, 8],

        [2, 4, 6]

    ];

    return winningCombinations.some(combination =>

        combination.every(index => board[index] === player)

    );

}

Игра заканчивается и игровое поле обновляется, когда поля все заполнены и нет победителя – выводится сообщение Ничья! посредством alert(); когда найден победитель – выводится сообщение “Выиграл игрок Х/O” посредством alert(); или нажата кнопка “Сбросить”.

# СПИСОК ИСТОЧНИКОВ ИНФОРМАЦИИ

**1. Электронные ресурсы**

1. Сайт

Электронный учебник по HTML и CSS [Электронный ресурс]: офиц. сайт. <http://htmlbook.ru/html5> (Дата обращения: 26.06.2023).

2. Сайт

Официальная документация Mozilla [Электронный ресурс]: офиц. сайт. <https://developer.mozilla.org/ru/> (Дата обращения: 26.06.2023).

3. Сайт

Различные разделы с вопросами по игре Tic-Tac-Toe.

<https://stackoverflow.com/> (Дата обращения: 26.06.2023).

4. Сайт

Различные разделы с вопросами по игре Крестики-Нолики

<https://qna.habr.com/> (Дата обращения: 26.06.2023)

5. Сайт

Различные разделы с вопросами по игре Крестики-Нолики

<https://www.geeksforgeeks.org> (Дата обращения: 26.06.2023)

6. Сайт

Различные разделы с вопросами по игре “Угадай число”

<https://www.cyberforum.ru> (Дата обращения: 27.06.2023)

**2. Примеры реализаций игровых порталов**

1. Сайт

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

2. Сайт

<https://kumpulan.games/>

3. Сайт

<https://vkplay.ru/>