МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Р.Е. АЛЕКСЕЕВА»

(НГТУ)

|  |  |  |
| --- | --- | --- |
| Институт | Институт радиоэлектроники и информационных технологий | |
| Направление подготовки | 09.03.02 Информационные системы и технологии | |
| Направленность (профиль) образовательной программы | | Информационные технологии в дизайне |
| Кафедра | Графические информационные системы | |

**ОТЧЕТ**

**по прохождению Ознакомительной практики**

|  |  |
| --- | --- |
| Выполнил Ромашкин Владислав Андреевич |  |
| Студент гр. 23-ИСТ-4-1 |  |
| Руководитель практики от кафедры Филинских А.Д. | |
| Степень, звание | к.т.н., доцент |
|  |  |
|  |  |
| Отчет защищен с оценкой | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | « » июля 2024 г. |

Нижний Новгород

2024 год

Оглавление

[ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 3](#_Toc140640690)

[Введение. 6](#_Toc140640691)

[Разработка веб-страницы с использованием языка программирования JavaScript. 7](#_Toc140640692)

[Веб-страница. 7](#_Toc140640693)

[JavaScript. 8](#_Toc140640694)

[HTML. 9](#_Toc140640695)

[CSS. 10](#_Toc140640696)

[Идея веб-страницы Таймер. 11](#_Toc140640697)

[Создание макета веб-страницы. 11](#_Toc140640698)

[Верстка. 13](#_Toc140640700)

[Инструменты для верстки и разработки. 14](#_Toc140640701)

[Visual Studio Code. 14](#_Toc140640702)

[Программирование. 16](#_Toc140640703)

[Структура Таймера. 17](#_Toc140640704)

[Тестирование веб-страницы. 18](#_Toc140640705)

[Заключение. 19](#_Toc140640706)

[Список литературы 21](#_Toc140640707)

[Приложение. 22](#_Toc140640708)

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Р.Е. АЛЕКСЕЕВА»

(НГТУ)

Кафедра Графические информационные системы

|  |  |
| --- | --- |
|  | Утверждаю: |
|  | Заведующий кафедрой |
|  | А.Д. Филинских |
|  | «06» июля 2024 г. |

# ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

на Учебную практику (Ознакомительная практика)

Студента гр. 23-ИСТ-4-1 Ромашкина Владислава Андреевича

Направление подготовки/специальность: 09.03.02 Информационные системы и технологии

Профиль подготовки: Информационные технологии в дизайне

Место прохождения практики НГТУ им. Р.Е. Алексеева, кафедра «Графические информационные системы»

Время прохождения практики

Дата начала практики «06» июля 2024 г.

Дата окончания практики «19» июля 2024 г.

**Тема индивидуального задания:**

**Таймер обратного отсчёта**

Содержание практики

*Во время прохождения практики студент обязан:*

**Ознакомиться:** с программой прохождения практики и основными задачами, выполняемыми в процессе профессиональной деятельности.

**Изучить:** области применения языка JavaScript, а также принципы применения практических навыков, связанных с будущей профессиональной деятельностью.

**Выполнить следующие виды работ по приобретению практических навыков:**

1. Создание макета сайта.
2. Разработка сайта.
3. Программирование.
4. Тестирование сайта.
5. Оформление отчета по практике.

Должность на практике практикант

Планируемые результаты обучения при прохождении практики

| **Планируемые результаты освоения образовательной программы** | **Планируемые результаты обучения при прохождении практики** | | | |
| --- | --- | --- | --- | --- |
| **Код компетенции** | **Знать** | **Уметь** | **Владеть** | |
| ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности | * основы математики, физики, вычислительной техники и программирования; | * решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования; | * навыками теоретического и экспериментального исследования объектов профессиональной деятельности; |
| ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; | * принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | - решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | * навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности | |
| ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий; | |  | | --- | | * основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий | | * применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ | * навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач | |

Руководитель практики от кафедры

Зав. каф. ГИС к.т.н., доцент Филинских А.Д.

(ученое звание и степень)(подпись) Ф.И.О.

Задание на практику получил:

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_ Ромашкин В.А.

(подпись) Ф.И.О.

«06» июля 2024 г.

**РАБОЧИЙ ГРАФИК (ПЛАН)**

**ПРОВЕДЕНИЯ Учебной практики**

Студента гр. 23-ИСТ-4-1 Ромашкина Владислава Андреевича

| №№ п/п | Разделы (этапы) практики | Сроки выполнения  с «06» июля 2024 г. по «19» июля 2024 г. | Отметка о выполнении (подпись руководителя практики) |
| --- | --- | --- | --- |
| 1 | **Подготовительный (организационный) этап** | с «06» июля 2024 г. по «09» июля 2024 г. |  |
| 1.1 | Определение базы прохождения практики | 06 июля 2024 г. |  |
| 1.2 | Организационное собрание для разъяснения целей, задач, содержания и порядка прохождения практики | 09 июля 2024 г. |  |
| 1.3 | Получение индивидуального задания | 09 июля 2024 г. |  |
| 2 | **Выполнение индивидуального задания:** | с «09» июля 2024 г. по «17» июля 2024 г. |  |
| 2.1 | Создание макета веб-страницы | 13 июля 2024 г. |  |
| 2.2 | Составление CSS таблицы для веб-страницы | 14 июля 2024 г. |  |
| 2.3 | Написание функций JavaScript | 15 июля 2024 г |  |
| 2.4 | Создание итогового веб-страницы | 16 июля 2024 г |  |
| 2.5 | Проверка на ошибки, их исправление | 17 июня 2024 г |  |
| 3 | Заключительный этап | с «18» июля 2024 г.  по «19» июля 2024 г. |  |
| 3.1 | *Написание и оформление отчета по практике* | 18 июля 2024 г. |  |
| 3.2 | *Защита отчета по практике* | 19 июля 2024 г. |  |

Руководитель практики от кафедры

Зав. каф. ГИС, доцент, к.т.н. Филинских А.Д.

*(ученые звание и степень) (подпись)* Ф.И.О.

# Введение.

**Цель прохождения практики:** формирование профессиональных навыков и умений в ходе изучения языка JavaScript.

**Задачи поставленные при прохождении практики:**

* Ознакомиться с программой прохождения практики и основными задачами, выполняемыми в процессе профессиональной деятельности.
* Изучить современные методы разработки веб-страниц и сайтов.
* Изучить принципы применения практических навыков, связанных с будущей профессиональной деятельностью.
* Разработать свою веб-страницу.

# Разработка веб-страницы с использованием языка программирования JavaScript.

### Веб-страница.

**Веб-страница** — документ или информационный ресурс Всемирной паутины, доступ к которому осуществляется с помощью веб-браузера. Сеть не является Интернетом, она лишь использует Интернет как среду передачи информации и данных.

Типичная веб-страница представляет собой текстовый файл в формате HTML, который может содержать ссылки на файлы в других форматах (текст, графические изображения, видео, аудио, мультимедиа, апплеты, прикладные программы, базы данных, веб-службы и прочее), а также гиперссылки для быстрого перехода на другие веб-страницы или доступа к ссылочным файлам.

Многие современные браузеры позволяют просмотр содержания ссылочных файлов непосредственно на веб-странице, содержащей ссылку на данный файл. Современные браузеры также позволяют прямой просмотр содержания файлов определённых форматов, в отрыве от веб-страницы, которая на них ссылается.

### 

### JavaScript.

JavaScript – это язык программирования, который используют разработчики для создания интерактивных веб-страниц и веб-приложений.



Рисунок 1. Логотип JavaScript

Функции JavaScript могут улучшить удобство взаимодействия пользователя с веб-сайтом: от обновления ленты новостей в социальных сетях и до отображения анимации и интерактивных карт. JavaScript является языком программирования при разработке скриптов для выполнения на стороне клиента.

Он позволяет создавать динамические элементы на страницах, обрабатывать события, взаимодействовать с пользователями и многое другое. JavaScript является одним из трех языков, используемых в веб-разработке, вместе с HTML и CSS. Он является базовой технологией во всемирной сети Интернет и используется для выполнения скриптов на стороне клиента.

Преимущества языка JavaScript:

* Незаменимость в веб-разработке.
* Простота освоения и использования
* Независимость от платформы
* Снижение нагрузки на сервер
* Улучшение пользовательского интерфейса
* Поддержка параллелизма

### HTML.

HTML — это язык разметки гипертекстовых документов. Он нужен, чтобы отображать в браузере специальным образом отформатированный документ с множеством вложенных элементов: заголовками, абзацами, списками, гиперссылками, расположением изображений, видео и аудио.



Рисунок 2. Логотип HTML

HTML используется для создания веб-страниц и определения их структуры и содержания. HTML является текстовым языком, который интерпретируется веб-браузерами. Он может быть использован в сочетании с другими технологиями, такими как CSS (каскадные таблицы стилей) и JavaScript, для создания визуально привлекательных и интерактивных веб-страниц. HTML прост в изучении и использовании, что делает его идеальным выбором для начинающих, которые хотят создавать веб-страницы.

### CSS.

CSS – это формальный язык, служащий для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML). Название происходит от английского Cascading Style Sheets, что означает «каскадные таблицы стилей».



Рисунок 3. Логотип CSS

Назначение CSS – отделять то, что задает внешний вид страницы, от ее содержания. Если документ создан только с использованием HTML, то в нем определяется не только каждый элемент, но и способ его отображения (цвет, шрифт, положение блока и т. д.). Если же подключены каскадные таблицы стилей, то HTML описывает только очередность объектов. А за все их свойства отвечает CSS. В HTML достаточно прописывать класс, не перечисляя все стили каждый раз.

Такая технология:

* + обеспечивает относительно простую и быструю разработку, потому что однажды созданное оформление можно применять ко многим страницам;
  + повышает гибкость и удобство редактирования – достаточно внести правку в CSS, чтобы оформление изменилось везде;
  + ускоряет время загрузки, потому что CSS может кэшироваться при первом открытии, а в последующих считываются только структура и данные;

# Идея веб-страницы Таймер.

В качестве темы для своего индивидуального задания, я выбрал разработку веб-страницы таймера обратного отсчёта, так как посчитал эту тему не слишком сложной для реализации и понимания.

Данный таймер можно использовать в быту для отсчёта времени до нужных событий или для напоминания для себя в жизни или за компьютером.

# Создание макета веб-страницы.

В целом, создание прототипа является важным этапом разработки веб-страницы, который позволяет определить её основные характеристики и убедиться в удобстве использования. Для создания прототипа своей веб-страницы я выбрал онлайн-сервис Figma.

**Figma** – это инструмент для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени. Figma используют для создания баннеров, схем, графических рисунков, логотипов и, конечно, веб-дизайна.

Figma предоставляет широкий набор функций, которые делают процесс дизайна более эффективным и удобным. В нем можно создавать и редактировать векторные элементы, использовать различные шрифты и стили, добавлять интерактивность к макетам, а также создавать анимации и переходы между страницами.

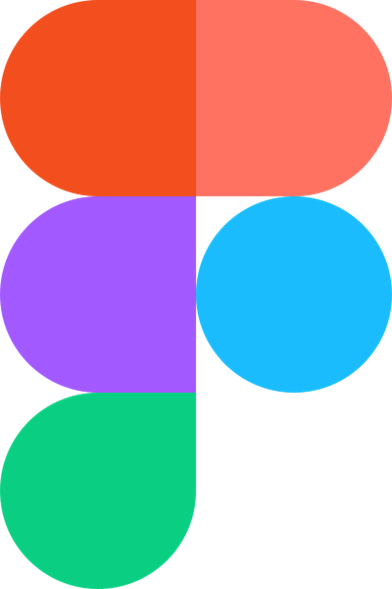


Рисунок 4. Логотип Figma

В целом, Figma является мощным и гибким инструментом для дизайна интерфейсов, который позволяет создавать профессиональные макеты и прототипы сайтов и веб-приложений.

При создании прототипа своей веб-страницы (рисунок 7) я определился с цветовым решением страницы и постарался обеспечить удобство его использования. Также я определил основные сценарии использования приложения.

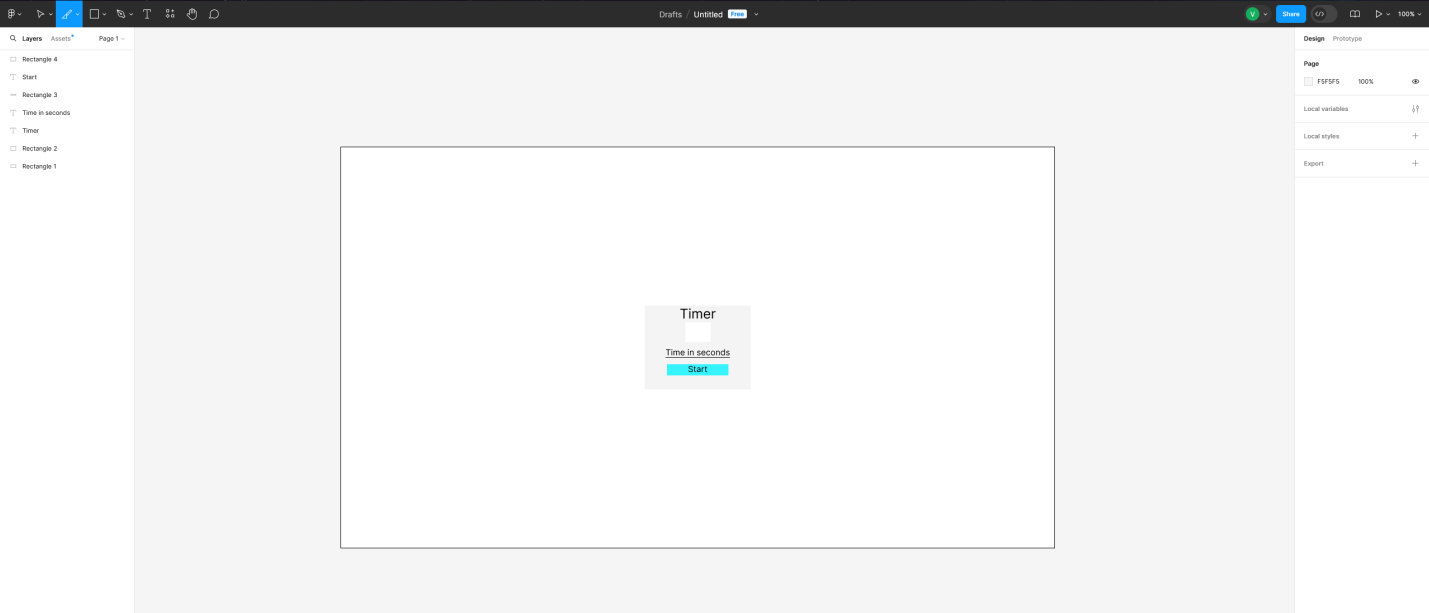


Рисунок 5. Прототип веб-страницы в Figma

На главном экране моей веб-страницы необходимо расположить: отображаемое время, поле ввода времени в секундах и кнопку старта таймера

# Верстка.

Верстка — это описание визуальной части сайта с помощью гипертекстового документа на основе HTML-разметки. Процесс создания веб-страницы или интерфейса, который включает в себя организацию и структурирование содержимого, размещение элементов на странице и определение их внешнего вида с помощью языков разметки и стилей.

Верстка сайта/веб-приложения — один из наиболее важных этапов создания интернет-ресурса. От нее зависит:

* + скорость загрузки сайта;
  + корректность его отображения в браузере;
  + соответствие стандартам HTML и требованиям поисковых систем;
  + адаптивность под разные устройства пользователей (планшет, смартфон, ноутбук).

Верстка относится к сфере frontend-разработки. Она помогает правильно расположить элементы на странице и сделать так, чтобы работать с ними было максимально удобно.

Основными языками разметки, используемыми для верстки веб-страниц, являются HTML (HyperText Markup Language) и XHTML (eXtensible HyperText Markup Language). HTML определяет структуру и содержимое страницы, такие как заголовки, абзацы, списки, изображения и ссылки. XHTML является расширением HTML и следует более строгим правилам разметки.

Для определения внешнего вида элементов на странице используются каскадные таблицы стилей (CSS - Cascading Style Sheets). CSS позволяет задавать цвета, шрифты, размеры, отступы, позиционирование и другие атрибуты элементов, чтобы создать желаемый дизайн и макет страницы.

### Инструменты для верстки и разработки.

Инструменты разработки - это программы, которые позволяют создавать, тестировать и отлаживать программное обеспечение.

В качестве инструмента разработки я выбрала Visual Studio Code.

Visual Studio Code (VS Code) — это редактор кода для разных языков программирования. Он относительно немного весит, гибкий и удобный. В нем можно писать, форматировать и редактировать код на разных языках.

VS Code позволяет легко писать, форматировать и редактировать код на разных языках. С его помощью можно быстро создать проект и структуру файлов в нем, он подсвечивает синтаксис кода и помогает автоматически править ошибки. В нем есть возможности для отладки и запуска кода на некоторых языках.

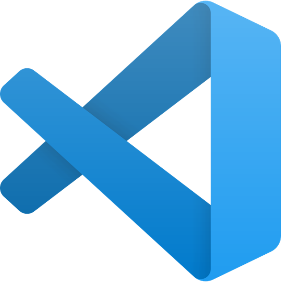


Рисунок 6. Логотип Visual Studio Code.

Возможности Visual Studio Code:

* + Написание и редактирование кода.
  + Автоматическое редактирование.
  + Поиск, вставка и другое.
  + Горячие клавиши.
  + Контроль версий.
  + Установка дополнений.
  + Запуск кода.
  + Помощь в отладке.

Интерфейс готовой веб-страницы.

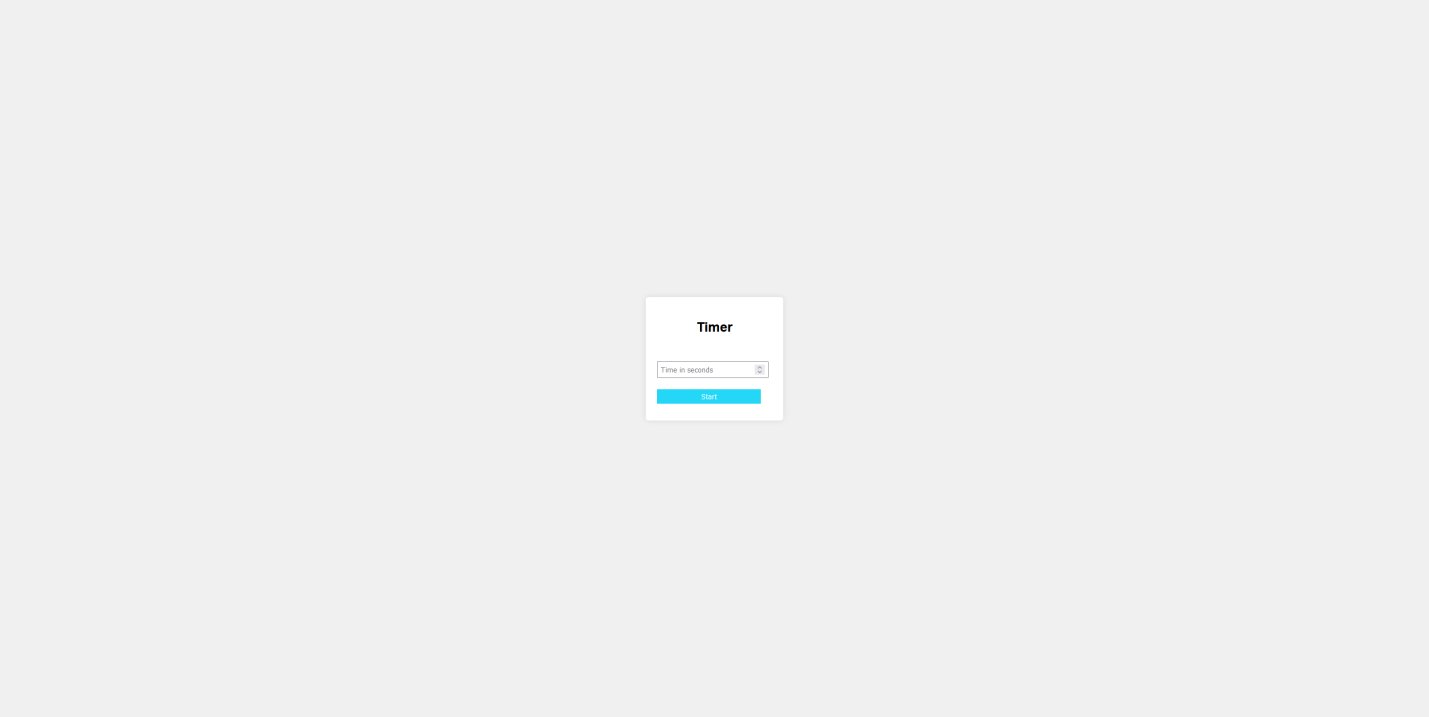


Рисунок 7. Веб-страница таймера

# 

# Программирование.

Программирование – это процесс создания компьютерных программ на особом языке, превращающий алгоритмы и структуры данных в некоторую инструкцию, которую может выполнить компьютер. В настоящее время существует большое количество различных языков программирования, которые дают возможность разными способами записать данные и создать определенные процессы.

Программирование включает в себя несколько этапов, таких как проектирование, написание кода, тестирование и отладка. В процессе программирования разработчик использует специальные инструменты, такие как текстовые редакторы, интегрированные среды разработки и компиляторы.

Программирование используется для создания различных приложений и программ, включая мобильные приложения, игры, веб-сайты и многое другое. Это очень важный навык в современном мире, так как компьютеры и программы используются в различных областях, включая науку, бизнес и развлечения. Оно также используется для автоматизации задач и процессов в различных отраслях.



Рисунок 8.

### Структура Таймера.

Таймер представляет из себя веб-страницу, на которой расположены:

* + Таймер обратного отсчёта
  + Поле ввода времени
  + Кнопка запуска таймера

В поле ввода вводится число (числа меньше 0 не принимаются).

После нажатия кнопки запускается таймер.

Когда таймер достигает 0, на экран выводится уведомление о завершении его работы.

# Тестирование веб-страницы.

Тестирование сайта — один из важных жизненных этапов, после которого, предоставляется заказчику готовый проект без ошибок, с хорошей читабельностью, воспринимаемой легкостью, удобством и надежностью. Мы хотели бы рассказать о том какие этапы проходит сайт перед выходом в продакшн.

Тестирование — это отклонение фактического результата от ожидаемого, другими словами — это процесс поиска багов (ошибок).

Основные правила тестирования веб-сайтов — это шаги, которые показывают пользователю, насколько удобен и логичен будет проект, насколько просто и возможно найти ту или иную информацию. Хорошо ли воспринимается человеческому взгляду и правильно ли работает весь функционал данного сайта, который был поставлен по ТЗ или брифу — это основные показатели для тестировщика.

Вот некоторые основные типы тестирования, которые применяются для веб-страниц и сайтов:

1. Функциональное тестирование
2. Тестирование производительности
3. Юзабилити-тестирование
4. Тестирование пользовательского интерфейса
5. Тестирование безопасности
6. Тестирование совместимости

Тестированию своей веб-страницы я посвятил достаточно времени. Благодаря тестированию я смог выявить некоторые неочевидные проблемы в работе страницы, некоторые из которых удалось устранить. Без тестирования эти ошибки было бы в разы сложнее обнаружить.

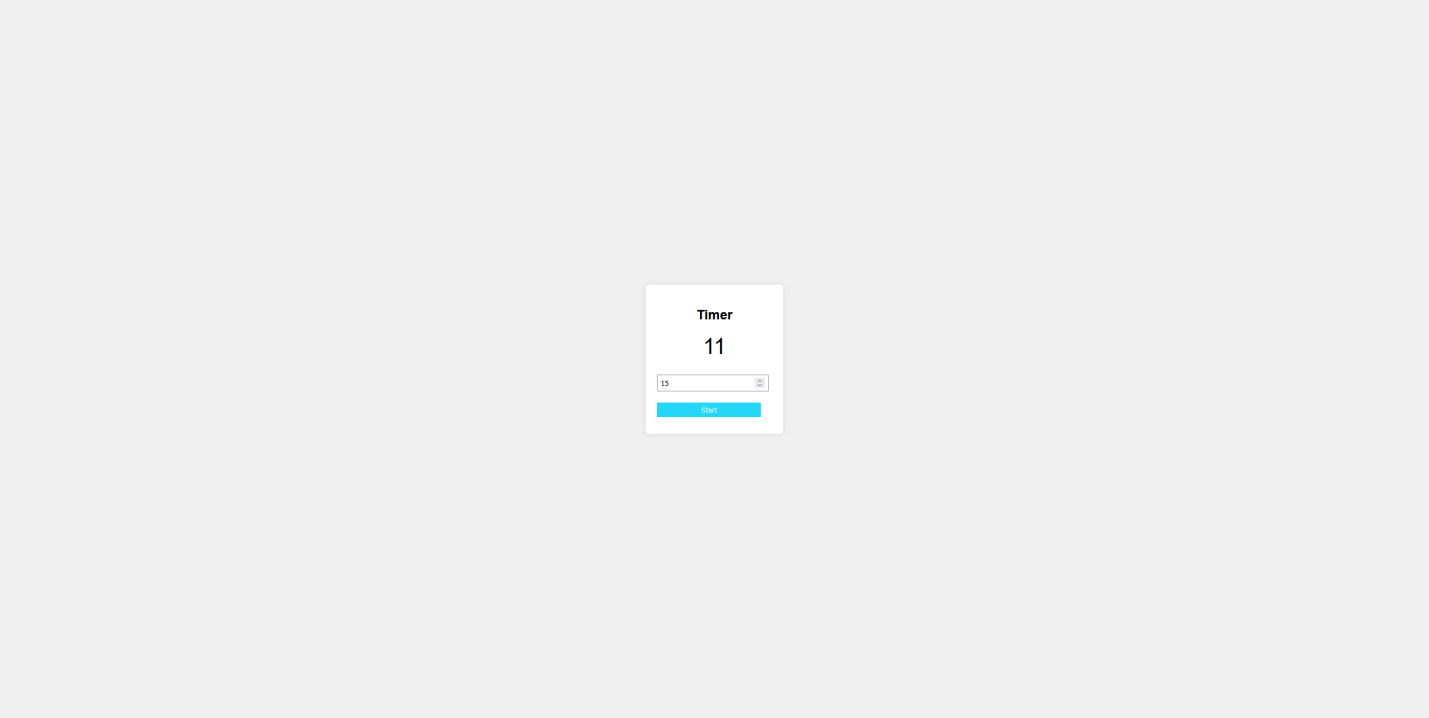


Рисунок 9. Тестирование страницы

# Заключение.

В процессе практики по JavaScript все поставленные в начале практики задачи были выполнены.

Мною были изучены современные методы разработки веб-страниц, сайтов и веб-приложений, инструменты, используемые в разработке. А также принципы применения практических навыков, связанных с будущей профессиональной деятельностью.

Практика по JavaScript позволила получить ценный опыт в разработке веб-страниц и сайтов. Она помогла улучшить навыки программирования, понимание языка JavaScript и его возможностей, а также понимание процесса разработки веб-страниц.

# Список литературы

Современный учебник JavaScript

<https://learn.javascript.ru/>

METANIT.COM

Сайт о программировании

<https://metanit.com/web/javascript/>

Figma Learn

<https://help.figma.com/hc/en-us>

htmlbook.ru

<http://htmlbook.ru/>

W3School

<https://www.w3schools.com/>

JavaScript.ru

<https://javascript.ru/>

HTML Academy

<https://htmlacademy.ru/>

# Приложение.

index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Timer</title>

    <link rel="stylesheet" href="css\CSS.css">

</head>

<body>

    <script src="js\startTimer.js"></script>

    <div class="container">

        <h2 style="text-align: center;">Timer</h2>

        <div class='time-block'></div><br/>

        <input type="number" min="1" class='time-input' placeholder="Time in seconds">

        <button class='time-button' ">Start</button>

    </div>

</body>

</html>

startTimer.js

document.addEventListener('DOMContentLoaded', function () {

  let timeInput = document.querySelector('.time-input');

  let timeButton = document.querySelector('.time-button')

  let timeBlock = document.querySelector('.time-block');

  function inputTime() {

    timeBlock.textContent = timeInput.value;

  }

  let timerID;

  function start() {

    clearInterval(timerID); // Сбрасываем таймер

    let time = parseInt(timeInput.value); // Запоминаем исходное значение

    timerID = setInterval(function() {

      const res = --time; // Каждый вызов уменьшаем

      timeBlock.textContent = res; // Выводим результат

      if (!res) {clearInterval(timerID);

                 alert("ВРЕМЯ ВЫШЛО!");};

    }, 1000)

  }

  timeButton.addEventListener('click', start);

  timeInput.addEventListener('input', inputTime);

});

CSS.css

body {

            font-family: Arial, sans-serif;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

            margin: 0;

            background-color: #f0f0f0;

        }

        .container {

            background-color: white;

            padding: 20px;

            border-radius: 5px;

            box-shadow: 0 0 10px rgba(0,0,0,0.1);

        }

        input, button {

            margin: 10px 0;

            padding: 5px;

            width: 90%;

        }

        button {

            background-color: #25d7f7;

            color: white;

            border: none;

            cursor: pointer;

        }

        button:hover {

            background-color: #22c7e4;

        }

        .time-block {

        font-size: 40px;

        text-align: center;

        }

        #result {

            margin-top: 20px;

            font-weight: bold;

        }