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 |  |
| Студент гр. 23-ИСТ-4-1 | Покатова Е.А. |
| Руководитель практики от кафедры Филинских А.Д. | |
| Степень, звание | к.т.н., доцент |
|  |  |
|  |  |
| Отчет защищен с оценкой | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | «19» июля 2024 г. |

Нижний Новгород

2024 год.

Оглавление

[ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 3](#_Toc172249454)

[Введение. 6](#_Toc172249455)

[Основная часть. Разработка веб-приложения “Таймер” на языке программирования JavaScript. 7](#_Toc172249456)

[Заключение 20](#_Toc172249457)

[Список используемой литературы 21](#_Toc172249458)

[Приложение 22](#_Toc172249459)

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 Покатовой Екатерины Александровны

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 до назначенной даты**

Содержание практики

*Во время прохождения практики студент обязан:*

**Ознакомиться:** с синтаксисом и функциями языков JavaScript и HTML.

**Изучить:** применения языков программирования JavaScript и HTML при создании сайтов.

**Выполнить следующие виды работ по приобретению практических навыков:** разработать дизайн сайта, разработать код сайта на HTML, прописать функцию работы таймера и сохранения времени на языке JavaScript.

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 \_\_\_\_\_\_\_\_\_\_\_\_\_ Покатова Е.А.

(подпись) (ФИО)

«06» июля 2024 г.

**РАБОЧИЙ ГРАФИК (ПЛАН)**

**ПРОВЕДЕНИЯ Учебной практики**

Студента гр.23-ИСТ-4-1 Ф.И.О. Покатовой Екатерины Александровны

| №№ п/п | Разделы (этапы) практики | Сроки выполнения  с «6» июля 2024 г. по «19» июля 2024 г. | Отметка о выполнении (подпись руководителя практики) |
| --- | --- | --- | --- |
| 1 | **Подготовительный (организационный) этап** | 06 июля 2024 г.  07 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 | 07 июля 2024 г. |  |
| 1.3 | Получение индивидуального задания | 07 июля 2024 г. |  |
| 2 | **Выполнение индивидуального задания:** | 08 июля 2024 г.  17 июля 2024 г. |  |
| 2.1 | Разработка дизайна сайта в Figma | 09 июля 2024 г. |  |
| 2.2 | Разработка сайта в соответствии с дизайном | 11 июля 2024 г. |  |
| 2.3 | Разработка функции работы сохранения даты на языке JavaScript. | 13 июля 2024 г. |  |
| 2.4 | Разработка функции работы таймера на языке JavaScript. | 15 июля 2024 г. |  |
| 2.5 | Доработка сайта и исправление ошибок | 17 июля 2024 г. |  |
| 2.6 |  |  |  |
| 2.6 |  |  |  |
| 2.6 |  |  |  |
| 2.7 |  |  |  |
| 2.8 |  |  |  |
| 2.9 |  |  |  |
| 3 | Заключительный этап | 18 июля 2024 г.  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

доцент, к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_ \_Филинских А.Д,\_\_\_\_\_\_\_\_\_

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

# Введение.

**Актуальность:**

Ежедневно каждый человек сталкивается с часами или календарем и их использованием. Однако бывает, что нужно вести счет дней и часов до определенной даты. Для этого и был создан сайт с таймером.

**Цель практики:**

Формирование профессиональных навыков и умений в ходе изучения языка JavaScript и HTML.

**Цель:**

Разработать сайт на языке HTML с таймером, функции которого прописаны JavaScript

**Задачи:**

1. Ознакомиться с программой прохождения практики и основными задачами, выполняемыми в процессе профессиональной деятельности.
2. Разработка дизайна сайта в Figma
3. Разработка сайта в соответствии с дизайном
4. Разработка функции работы сохранения даты на языке JavaScript.
5. Разработка функции работы таймера на языке JavaScript.
6. Доработка сайта и исправление ошибок
7. Написание и оформление отчета по практике
8. Защита отчета по практике

**Список компетенций:**

1. Навыки использования Figma для создания дизайна;
2. Навык разработки сайтов на языке HTML;
3. Навык разработки функций и их использования на языке JavaScript;
4. Навык работы с системой контроля версий Git;

# Основная часть. Разработка веб-приложения “Таймер” на языке программирования JavaScript.

**Веб-приложение** – это компьютерная программа, которая запускается прямо в веб-браузере. Для работы не требуется установка на компьютер, доступ к нему можно получить с любого устройства, имеющего веб-браузер, при условии, что оно подключено к интернету.

Веб-приложение позволяет пользователям взаимодействовать с ним — вводить данные и получать результаты.

Веб-приложения построены на основе современных технологий, таких как HTML5, CSS3, JavaScript, и фреймворках, такие как Angular, React или Vue.js.

Веб-приложения имеют несколько преимуществ перед мобильными приложениями:

1. Универсальность: они всегда доступны, вне зависимости от вашего местоположения и используемого устройства. Для начала использования вам не нужно загружать и устанавливать приложение.
2. Скорость работы: они, как правило, быстрее мобильных приложений.
3. Экономия места: не занимают место на вашем устройстве, что может быть особенно важно для пользователей смартфонов с ограниченным объемом памяти.
4. Автоматическое обновление контента: веб-приложения автоматически обновляют информацию, как только она становится доступной, без необходимости ручного обновления приложения. [1]

Для разработки веб-сайта были выбраны следующие языки: HTML5(+ CSS) и JavaScript.

HTML (HyperText Markup Language) – язык разметки гипертекста представляет собой формат документов, включающий в себя следующие компоненты:

* сам текст
* теги (элементы) для выделения и структурирования текста

Преимущества HTML5:

* большое количество семантических элементов, таких как <header>, <nav>, <section>, <article> и другие, которые улучшают и упрощают структурирование контента на сайте
* поддержка нативных элементов для аудио и видео с возможностью воспроизведения без сторонних плагинов
* поддержка локального хранилища (LocalStorage и SessionStorage) для сохранения данных на стороне клиента
* наличие элемента <canvas> для создания графики и рендеринга с помощью JavaScript
* встроенная поддержка валидации форм с использованием атрибутов required, pattern и других
* большее внимание к семантике элементов, что способствует лучшей индексации поисковыми системами и доступности
* поддержка более строгой обработки ошибок, что помогает избежать некорректного отображения страницы
* возможность определения географического положения пользователя через API геолокации
* встроенная поддержка оффлайн-режима с помощью Application Cache и Service Workers [2]

CSS (Cascading Style Sheets) – каскадные таблицы стилей.

CSS является отдельным языком для стилевой разметки веб-страниц. Если HTML отвечает за структуру документа, то CSS отвечает за его внешний вид. Вместо использования тегов для форматирования текста, CSS позволяет задавать стили для элементов HTML, что упрощает и улучшает структуру исходного кода [3].

Javascript – динамический скриптовый язык программирования высокого уровня. Он отличается мультипарадигменностью. Речь идет о поддержке функционального, императивного, событийно-ориентированного стилей

Javascript поддерживается всеми популярными браузерами, полностью интегрируем с вёрсткой страниц (HTML+CSS) и серверной частью (backend). [4]

Для адаптивной вёрстки был использован Bootstrap v.5.0, позволяющий использовать веб-ресурс на любых устройствах.

Преимущества Bootstrap v.5.0:

* Уменьшение количества времени, затрачиваемого на разработку.
* Использование фреймворков и библиотек значительно облегчает работу разработчикам и позволяет разрабатывать проекты быстрее. Bootstrap дает готовые решения: их использование позволяет сократить время, затрачиваемое на создание макета сайта.
* Адаптивность
* Bootstrap позволяет создавать адаптивные сайты. Дизайн сайта будет корректно отображаться на экранах устройств разных размеров вне зависимости от их диагонали.
* Кросс-браузерность.
* Сайты, сделанные с использованием Bootstrap, будут одинаково отображаться во всех современных браузерах.
* Легкость в использовании и быстрота в освоении.
* Bootstrap легко использовать в разработке, в нем легко разобраться.

А те, кто еще не знаком с Bootstrap, смогут быстро научиться работать с этим инструментом. И дело не только в его простоте, но и в большом

* количестве уроков и инструкций, которые можно найти на просторах сети.
* Понятный код
* Bootstrap позволяет писать качественный и понятный код, который легко поймет другой разработчик. Это значительно упрощает разработку в команде. [5]

В качестве программы для разработки макета сайта был выбран графический редактор Figma.

Преимущества Figma:

* Много режимов редактирования
* Наличие персонального облака. Все изменения сохраняются в программе автоматически
* Наличие компонентов. Элементы пользовательского интерфейса могут принимать любые стили и подстраиваться под изменения непосредственно в макете
* История версий
* Фреймы.. Фреймы легко масштабируются, и могут использовать в других фреймах. Для них были созданы определенные размеры устройств
* Наличие сетки. Специальная панель Grid Layout создает сетки любого типа
* Комментирование макетов. (проверка макета сайта консультантом КР)
* Доступность

В качестве инструмента разработки был выбран редактор Visual Studio Code.

Visual Studio Code (VS Code) — это редактор кода для разных языков программирования.

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.

Преимущества Visual Studio Code:

* **Простота освоения.** У редактора интуитивно понятный интерфейс, его легко установить и настроить.
* **Малый вес.** По сравнению с IDE редактор быстро работает и мало весит, не так требователен к ресурсам компьютера. При этом он довольно функционален, а огромное количество расширений позволяет добавлять в него новые возможности.
* **Кроссплатформенность.** Visual Studio Code есть для всех популярных операционных систем.
* **Поддержка разных языков.** Есть инструменты, которые предназначены специально для какого-то языка, например Python или C++. Visual Studio Code поддерживает огромное количество языков программирования, и они легко подключаются.
* **Гибкость.** У VS Code множество расширений и удобных функций. Помимо вещей, стандартных для редактора кода, вроде подсветки синтаксиса, VS Code имеет еще несколько дополнительных функций.

Тестирование и редактирование сайта были проведены в браузереYandex версии 23.11.2.771.

**Разработка макета в Figma**

Разработка дизайна сайта была проведена в бесплатном интернет-редакторе Figma.

Для того, чтобы начать создание макета сайта, нужно создать Frame и разделить его на 12 частей, чтобы создать основу для размещения объектов сайта при помощи Layout grid.

Header.

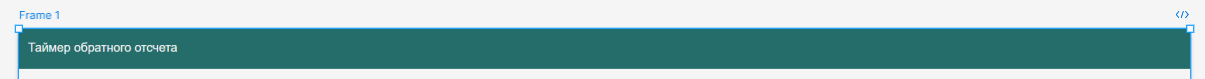


Рисунок 1 – Header

В header сайта представлен текстовый логотип “Таймер обратного отсчета”.

Footer.



Рисунок 2 – Footer

В footer сайта представлена почта создателя веб-ресурса.

Основная часть.

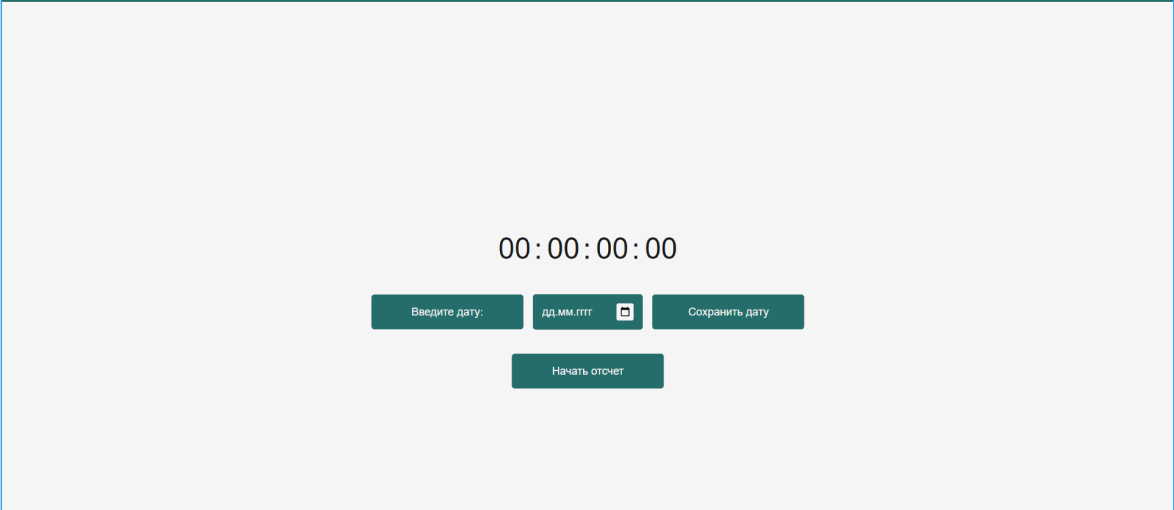


Рисунок 3 – Основная часть сайта

**Программная реализация**

Основные теги, которые присутствуют веб-странице:

* **<!DOCTYPE html>** – объявляет, что документ является документов HTML5, помогает браузерам правильно отображать веб-страницы.
* **<html>** – является корневым элементом HTML-страницы, контейнером, который заключает в себе все содержимое веб-страницы.
* **<head>** – содержит метаинформацию о HTML-странице. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных.
* **<body>** – определяет тело документа и является контейнером для всего видимого содержимого (заголовки, абзацы, изображения, гиперссылки, таблицы, списки и т. д.)

В парном теге <head> прописываем информацию о HTML-странице, а именно: теги <meta>, которые указывают на контроль масштаба страницы, а также кодировку; тег <title>, указывающий на заголовок HTML-страницы(он будет изменяться на каждой следующей странице); подключаем Bootstrap v.5; прописываем использование каскадных таблиц стилей (CSS). Закрываем тег </head>.

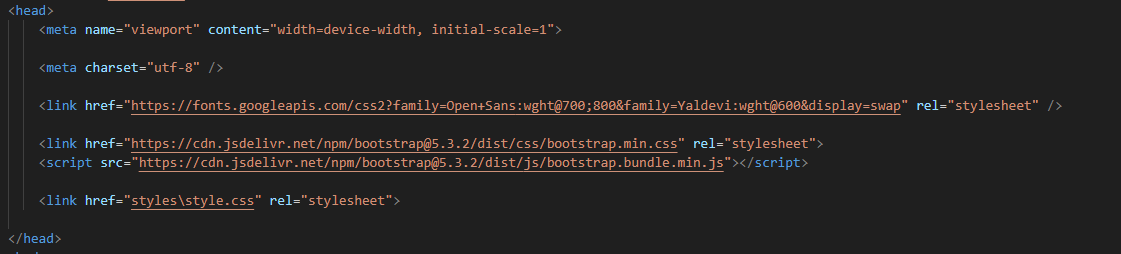
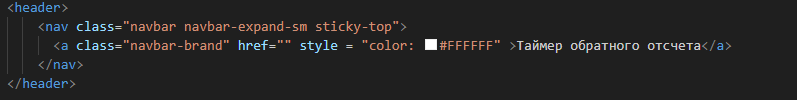


Рисунок 4 – Код(1)

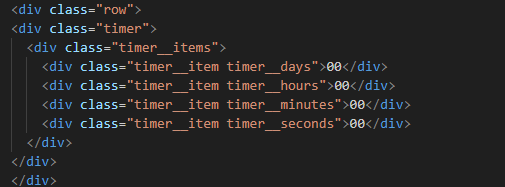


Рисунок 5 – Подключение JavaScript

Создаем Header при помощи класса navbar.

  
Рисунок 6 – Создание Header

Создаем таймер при помощи класса timer.

  
Рисунок 7 – Создание таймера

Создание формы ввода даты и кнопок. При нажатии кнопок вызываем функции, которые прописаны в файле func.js.

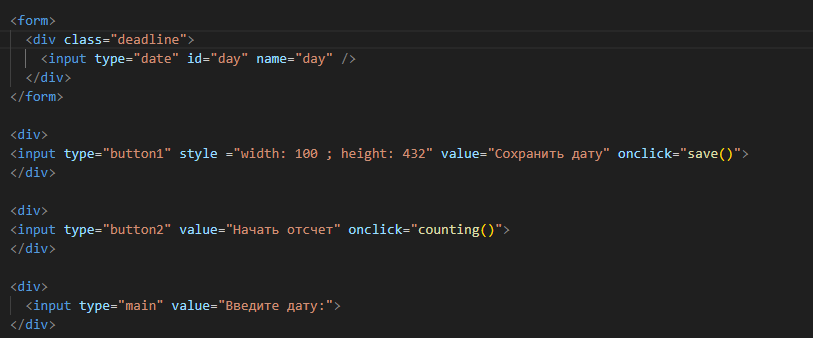


Рисунок 8 – Создание формы выбора даты и кнопок

Далее для каждого класса прописываем стили в файле style.css.

Указываем шрифты и Header, Footer.

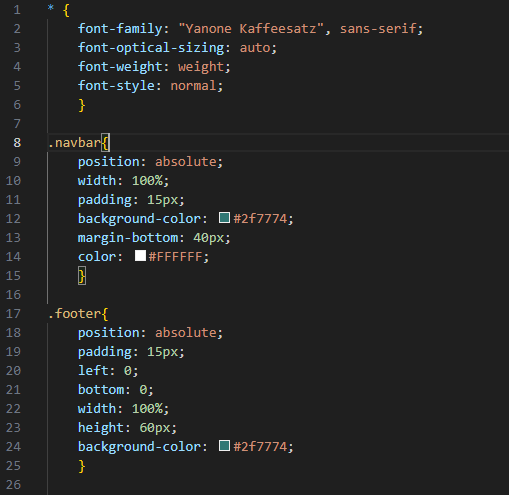
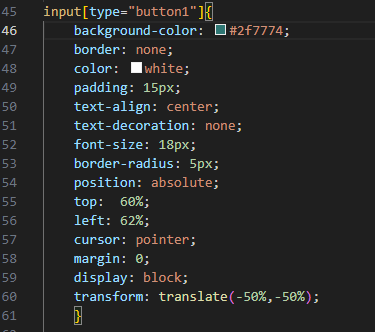
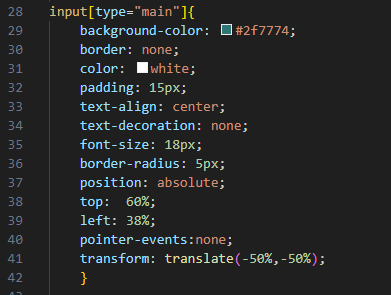
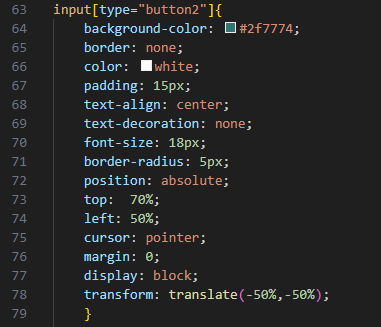


Рисунок 9 – Стили Header и Footer

Прописываем стили кнопок, а именно их цвет, расположение, внутренние и внешние отступы, скругление, контур, отображение курсора и размер текста.





Рисунки 10, 11, 12 – Стили кнопок

В стилях формы указываем ее расположение, цвет, отступы, размер текста, отображение курсора, скругление и контур.

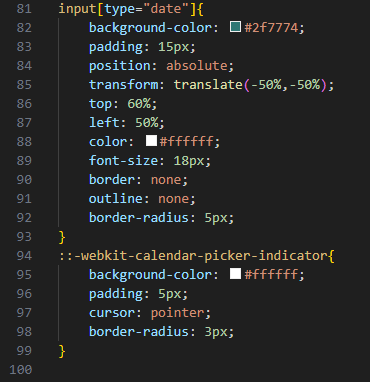


Рисунок 13 – Стили для формы выбора даты

Далее прописываем все стили для самого таймера: размер цифр, расположение, подписи, отступы между полями таймера,

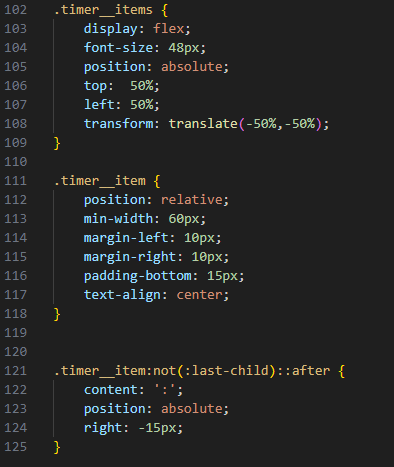


Рисунок 14 – Стили для таймера

Прописываем функции работы кнопок, а также работы самого таймера в файле func.js.

1. Сохранение выбранной даты.

Вводим переменную deadline, куда будем сохранять выбранную пользователем дату.

При помощи id считываем значение элемента с веб-страницы. Присваиваем переменной deadline значение новой даты. Если дата корректная, то выводим пользователю уведомление, что сохранение прошло успешно, иначе – просим заново ввести дату.

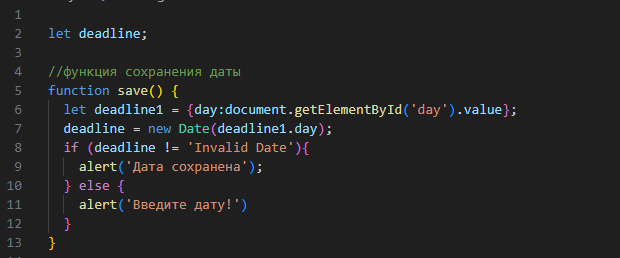


Рисунок 15 – Сохранение даты

2. Функция работы таймера.

Если пользователь не сохранил дату, но хочет начать отсчет, выводим ему уведомление с просьбой ввести дату.

Высчитываем разницу между выбранной датой и датой, установленной на данный момент на компьютере.

Если пользователь выбрал дату, которая уже прошла, выводим уведомление с просьбой изменения даты.

В константы $days, $hours, $minutes, $seconds сохраняем значения, которые на данный момент принимает таймер.

Вызываем функцию countdownTimer(). Далее вызываем ее каждую секунду для изменения времени.



Рисунок 16 – Начало работы таймера

При изменении времени также изменяться будут подписи временных единиц. Пропишем для этого функцию.

Функция проверяет последнюю цифру у числа и исходя из этого назначает верную форму слова. Сначала функция находит остатки от деления числа на 100 и 10. В 46 строке прописано, что если остаток при делении числа на 100 в промежутке от 11 до 19, то возвращается вторая форма существительного (дней, часов, минут, секунд). Если остаток при делении числа на 10 в промежутке от 1 до 5, то возвращается 1 форма существительного (дня, часа, минуты, секунды). Если же последняя цифра равняется 1, то возвращается 0 форма существительного (день, час, минута, секунда). Во всех же остальных случаях возвращается также 2 форма.

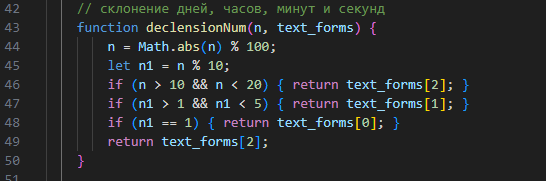


Рисунок 17 – Склонение существительных

Функция изменения значений чисел.

Высчитываем разницу между датами. Если она меньше или равна 0, то останавливаем таймер. Высчитываем константы days, hours, minutes, seconds: если разница между датами больше 0, тогда переводим значение из миллисекунд в нужную нам временную единицу при помощи деления и округления. Если разница между датами неположительная, то возвращаем 0.

Далее полученные значения присваиваем элементам на веб-странице. Если значение меньше 10, то перед ним дописываем 0, в другом случае возвращаем значение таким, какое оно есть.

Следующим шагом изменяем подписи временных единиц, используя функцию declensionNum(n, text\_forms), упомянутую ранее.

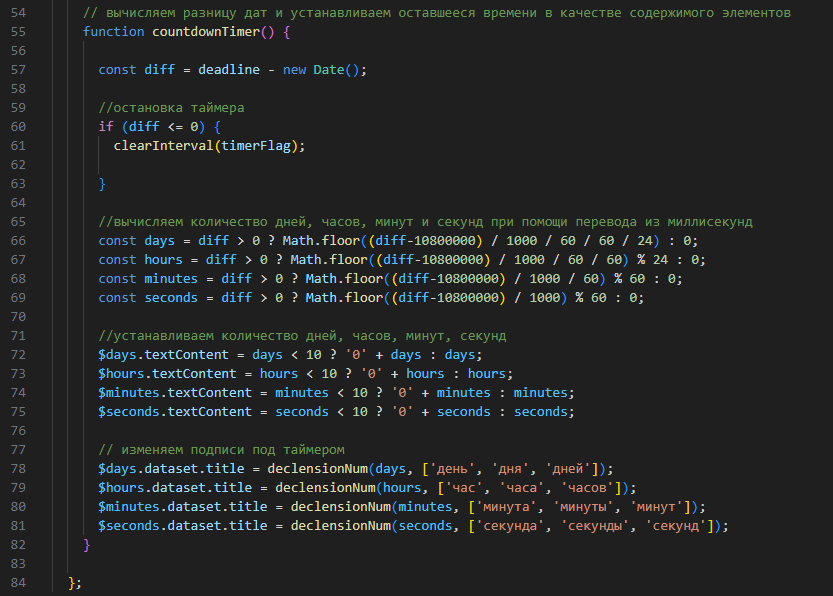


Рисунок 18 – Изменение оставшегося времени

# Заключение

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

Мной были изучены инструменты работы в Figma, функции и возможности JavaScript и HTML.

Благодаря практике в JavaScript был получен ценный опыт в создании веб-приложений, усовершенствованы программистские навыки, развито понимание языка JavaScript и его возможностей, а также улучшено понимание процесса разработки веб-приложений.

# Список используемой литературы

|  |  |
| --- | --- |
| [1] | «Что такое веб-приложение и чем оно отличается от сайта?,» [В Интернете]. Available: https://sibdev.pro/blog/articles/chto-takoe-veb-prilozhenie. |
| [2] | «HTML5: что это такое, возможности и отличия,» [В Интернете]. Available: https://blog.skillfactory.ru/glossary/html5/. |
| [3] | «CSS: что это такое – для чего нужны стили на сайте,» [В Интернете]. Available: https://www.sape.ru/glossary/css/. |
| [4] | «Язык программирования Javascript: особенности и преимущества,» [В Интернете]. Available: https://vc.ru/hr/145461-yazyk-programmirovaniya-javascript-osobennosti-i-preimushestva. |
| [5] | «Плюсы и минусы Bootstrap,» [В Интернете]. Available: https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1. |
| [6] | «W3 schools,» [В Интернете]. Available: https://www.w3schools.com/. |
| [7] | «Современный учебник JavaScript,» [В Интернете]. Available: https://learn.javascript.ru/. |
| [8] | «Таймер обратного отсчета,» [В Интернете]. Available: https://denis-creative.com/jstimer/. |
| [9] | Д. Дакетт, HTML и CSS. Разработка и дизайн веб-сайтов.. |
| [10] | «Уроки CSS,» [В Интернете]. Available: https://uroki-css.ru/. |

# Приложение

**Основной файл index.html**

<!DOCTYPE html>

<html>

    <script src="js/func.js"></script>

    <head>

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

        <meta charset="utf-8" />

        <link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@700;800&family=Yaldevi:wght@600&display=swap" rel="stylesheet" />

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

        <link href="styles\style.css" rel="stylesheet">

    </head>

    <body>

    <header>

        <nav class="navbar navbar-expand-sm sticky-top">

          <a class="navbar-brand" href="" style = "color: #FFFFFF" >Таймер обратного отсчета</a>

        </nav>

    </header>

    <div class="row">

    <div class="timer">

      <div class="timer\_\_items">

        <div class="timer\_\_item timer\_\_days">00</div>

        <div class="timer\_\_item timer\_\_hours">00</div>

        <div class="timer\_\_item timer\_\_minutes">00</div>

        <div class="timer\_\_item timer\_\_seconds">00</div>

      </div>

    </div>

    </div>

    <form>

      <div class="deadline">

        <input type="date" id="day" name="day" />

      </div>

    </form>

    <div>

    <input type="button1" style ="width: 100 ; height: 432" value="Сохранить дату" onclick="save()">

    </div>

    <div>

    <input type="button2" value="Начать отсчет" onclick="counting()">

    </div>

    <div>

      <input type="main" value="Введите дату:">

    </div>

  <div class="footer">

    <div class="row">

      <div class="col-sm-3" style="color: #FFFFFF;">

        <h5>eekaterinap.a@gmail.com</h5>

      </div>

  </div>

  </body>

</html>

**Файл со стилями style.css**

\* {

    font-family: "Yanone Kaffeesatz", sans-serif;

    font-optical-sizing: auto;

    font-weight: weight;

    font-style: normal;

    }

.navbar{

    position: absolute;

    width: 100%;

    padding: 15px;

    background-color: #2f7774;

    margin-bottom: 40px;

    color: #FFFFFF;

    }

.footer{

    position: absolute;

    padding: 15px;

    left: 0;

    bottom: 0;

    width: 100%;

    height: 60px;

    background-color: #2f7774;

    }

input[type="main"]{

    background-color: #2f7774;

    border: none;

    color: white;

    padding: 15px;

    text-align: center;

    text-decoration: none;

    font-size: 18px;

    border-radius: 5px;

    position: absolute;

    top:  60%;

    left: 38%;

    pointer-events:none;

    transform: translate(-50%,-50%);

    }

input[type="button1"]{

    background-color: #2f7774;

    border: none;

    color: white;

    padding: 15px;

    text-align: center;

    text-decoration: none;

    font-size: 18px;

    border-radius: 5px;

    position: absolute;

    top:  60%;

    left: 62%;

    cursor: pointer;

    margin: 0;

    display: block;

    transform: translate(-50%,-50%);

    }

input[type="button2"]{

    background-color: #2f7774;

    border: none;

    color: white;

    padding: 15px;

    text-align: center;

    text-decoration: none;

    font-size: 18px;

    border-radius: 5px;

    position: absolute;

    top:  70%;

    left: 50%;

    cursor: pointer;

    margin: 0;

    display: block;

    transform: translate(-50%,-50%);

    }

input[type="date"]{

    background-color: #2f7774;

    padding: 15px;

    position: absolute;

    transform: translate(-50%,-50%);

    top: 60%;

    left: 50%;

    color: #ffffff;

    font-size: 18px;

    border: none;

    outline: none;

    border-radius: 5px;

}

::-webkit-calendar-picker-indicator{

    background-color: #ffffff;

    padding: 5px;

    cursor: pointer;

    border-radius: 3px;

}

.timer\_\_items {

    display: flex;

    font-size: 48px;

    position: absolute;

    top:  50%;

    left: 50%;

    transform: translate(-50%,-50%);

}

.timer\_\_item {

    position: relative;

    min-width: 60px;

    margin-left: 10px;

    margin-right: 10px;

    padding-bottom: 15px;

    text-align: center;

}

.timer\_\_item:not(:last-child)::after {

    content: ':';

    position: absolute;

    right: -15px;

}

**Файл с функциями JavaScript func.js**

let deadline;

//функция сохранения даты

function save() {

  let deadline1 = {day:document.getElementById('day').value};

  deadline = new Date(deadline1.day);

  if (deadline != 'Invalid Date'){

    alert('Дата сохранена');

  } else {

    alert('Введите дату!')

  }

}

//функция отсчета времени

function counting() {

  let timerFlag = null;

  if (!deadline || deadline == 'undefined' || deadline == 'Invalid Date'){

    alert('Введите дату!')

  }

  diff0 = deadline - new Date();

  if (diff0 <= 0) {

    alert('Эта дата уже наступила. Выберите другую')

  }

  // получаем элементы, содержащие компоненты даты с сайта

  const $days = document.querySelector('.timer\_\_days');

  const $hours = document.querySelector('.timer\_\_hours');

  const $minutes = document.querySelector('.timer\_\_minutes');

  const $seconds = document.querySelector('.timer\_\_seconds');

  // вызываем функцию countdownTimer

  countdownTimer();

  // вызываем функцию countdownTimer каждую секунду

  timerFlag = setInterval(countdownTimer, 1000);

    // склонение дней, часов, минут и секунд

    function declensionNum(n, text\_forms) {

        n = Math.abs(n) % 100;

        let n1 = n % 10;

        if (n > 10 && n < 20) { return text\_forms[2]; }

        if (n1 > 1 && n1 < 5) { return text\_forms[1]; }

        if (n1 == 1) { return text\_forms[0]; }

        return text\_forms[2];

    }

    // вычисляем разницу дат и устанавливаем оставшееся времени в качестве содержимого элементов

    function countdownTimer() {

      const diff = deadline - new Date();

      //остановка таймера

      if (diff <= 0) {

        clearInterval(timerFlag);

      }

      //вычисляем количество дней, часов, минут и секунд при помощи перевода из миллисекунд

      const days = diff > 0 ? Math.floor(diff / 1000 / 60 / 60 / 24) : 0;

      const hours = diff > 0 ? Math.floor(diff / 1000 / 60 / 60) % 24 : 0;

      const minutes = diff > 0 ? Math.floor(diff / 1000 / 60) % 60 : 0;

      const seconds = diff > 0 ? Math.floor(diff / 1000) % 60 : 0;

      //устанавливаем количество дней, часов, минут, секунд

      $days.textContent = days < 10 ? '0' + days : days;

      $hours.textContent = hours < 10 ? '0' + hours : hours;

      $minutes.textContent = minutes < 10 ? '0' + minutes : minutes;

      $seconds.textContent = seconds < 10 ? '0' + seconds : seconds;

      // изменяем подписи под таймером

      $days.dataset.title = declensionNum(days, ['день', 'дня', 'дней']);

      $hours.dataset.title = declensionNum(hours, ['час', 'часа', 'часов']);

      $minutes.dataset.title = declensionNum(minutes, ['минута', 'минуты', 'минут']);

      $seconds.dataset.title = declensionNum(seconds, ['секунда', 'секунды', 'секунд']);

    }

  };