

**Государственное бюджетное общеобразовательное учреждение города
Москвы «Школа № 654 имени А.Д. Фридмана»**

Проект

ОНЛАЙН-КАЛЬКУЛЯТОР ДЛЯ ИНЖЕНЕРНЫХ РАСЧЁТОВ

Участники:

ученики ГБОУ Школы № 654
имени А.Д. Фридмана
Филенков Никита Сергеевич
ученик 10 «Б» класса
Панин Даниил Арифович
ученик 10 «Б» класса
Орлов Алексей Михайлович
ученик 10 «А» класса

Руководитель:

Гришина Арина Александровна

Москва, 2024

Оглавление

| | |
|--|----|
| Введение | 3 |
| 1. Основная часть | 4 |
| 1.1. Первый этап: | 5 |
| 1.2. Второй этап: | 5 |
| 1.3. Третий этап: | 6 |
| 1.4. Четвёртый этап: | 11 |
| 2. Функциональное и оценочное тестирование | 12 |
| Заключение..... | 15 |
| Список литературы | 16 |

Введение

Сегодня, в современном мире, инженерные онлайн-калькуляторы становятся все более актуальными. Это инновационные программы, которые облегчают жизнь инженерам, проектировщикам и другим специалистам в технической сфере, позволяя им быстро и точно производить необходимые расчеты.

Онлайн калькуляторы для инженерных расчетов предоставляют возможность быстро решать сложные математические задачи, которые раньше занимали много времени и усилий. Благодаря встроенному конвертеру физических величин можно легко переводить одни единицы измерения в другие, что значительно упрощает работу и предотвращает ошибки.

Одним из основных преимуществ онлайн калькуляторов является их доступность. Для их использования не нужно устанавливать специальное программное обеспечение, они доступны через любой браузер с доступом в интернет. Это позволяет инженерам использовать такие калькуляторы в любом месте и в любое время.

Таким образом, онлайн калькулятор для инженерных расчетов с встроенным конвертером физических величин становится неотъемлемой частью работы специалистов в технической сфере. Он облегчит процесс работы, увеличит производительность и точность результатов. Использование таких калькуляторов позволяет быть более эффективным и успешным в своей профессиональной деятельности.

Цель работы: создать сайт с инженерным калькулятором и конвертером физических величин.

Задачи:

- 1) Продумать функционал сайта.
- 2) Создать дизайн для калькулятора.
- 3) Написать код, в котором будет прописан весь функционал сайта.
- 4) Провести тестирование работоспособности функционирования сайта.

Конечным **продуктом** данного проекта является сайт с инженерный калькулятором и универсальным конвертером единиц измерения. Этот сайт станет отличным инструментом для облегчения сложных арифметических и инженерных вычислений.

Основная часть

Работа была выполнена с использованием ноутбука, и необходимых программных обеспечений: Figma, Visual studio code. Программа написана с помощью языка программирования Python, языка гипертекстовой разметки HTML, Java и фреймворка Django.

Первый этап:

Опираясь на задачи, поставленные перед этим проектом, первым этапом необходимо продумать функционал сайта, иерархию страниц и внедрить в онлайн-калькулятор, то, чего нет у аналогов. Так как калькулятор инженерный, было принято решение встроить конвертер физических величин. Сайт состоит из двух страниц: калькулятора и конвертера, переход между которыми осуществляется, по кнопке в верхнем левом углу страниц.

Второй этап:

Далее необходимо создать дизайн сайта. На рисунках 1 и 2 представлен готовый дизайн сайта, то есть то, как он будет выглядеть при его работе:

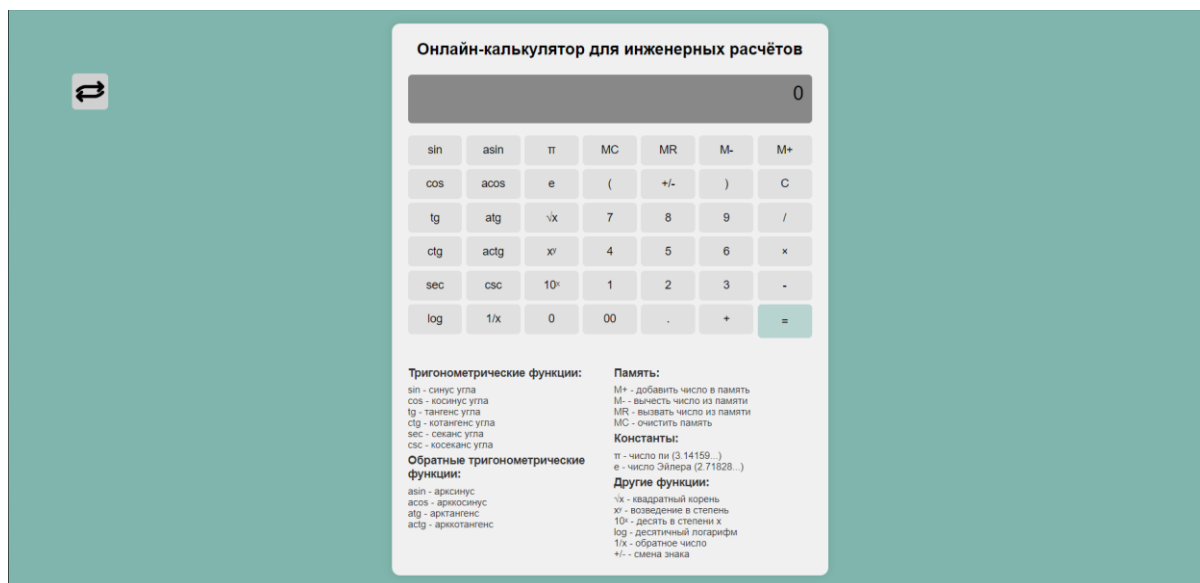


Рисунок 1 - Страница с калькулятором

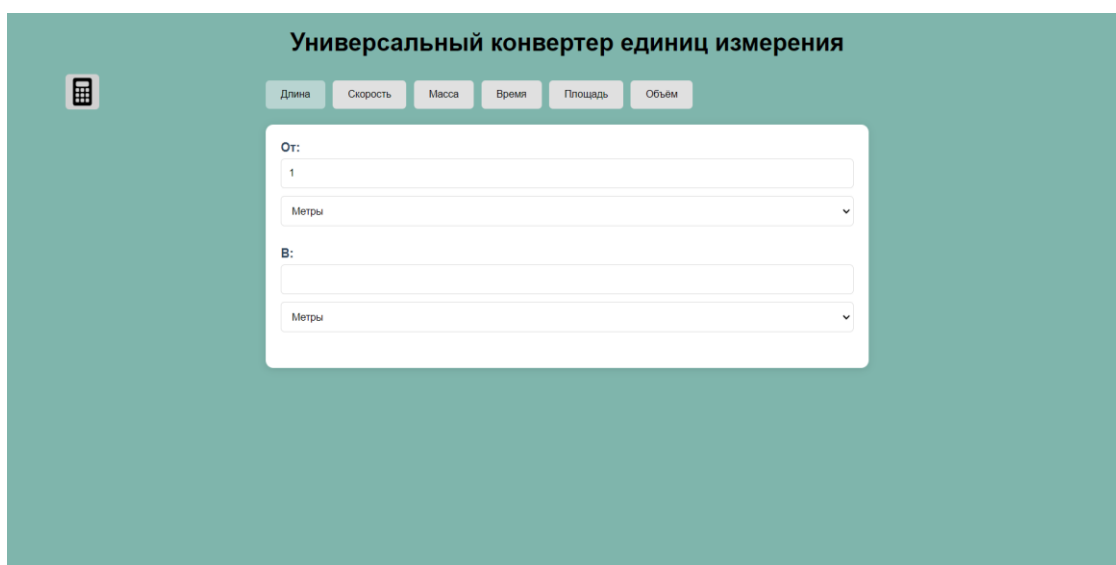


Рисунок 2 – Страница с конвертером единиц измерения

Третий этап:

Следующим шагом было написание кода. Программа написана на языке Python с помощью фреймворка Django, HTML и Java. Сначала нужно создать папку проекта, на рисунке 3 представлена структура проекта в Visual Studio Code:

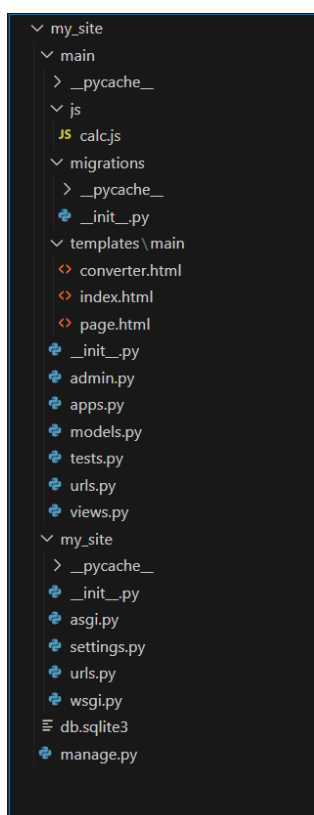


Рисунок 3 – Структура проекта в Visual Studio Code

Затем в каждом файле необходимо прописать несколько строчек кода для того, чтобы сайт функционировал. Рассмотрим файлы `views.py` и `urls.py`. Первый файл отвечает за представление сайта, а второй за маршрутизацию, то есть за переход между страницами сайта. Содержание этих файлов показаны на рисунках 4 и 5:

```
from django.shortcuts import render

def index(request):
    return render(request, 'main/index.html')

def page(request):
    return render(request, 'main/converter.html')
```

Рисунок 4 – Файл `views.py`, он определяет функции, которые получают запросы пользователей, обрабатывают их и возвращают ответ

```
from django.urls import path, include
from . import views

urlpatterns = [
    path('', views.index),
    path('converter', views.page)
]
```

Рисунок 5 – Файл `urls.py`, позволяет сопоставить маршруты с представлениями, которые будут обрабатывать запрос по этим маршрутам.

Теперь нужно перенести код из Figma в проект и отредактировать его, так чтобы сайт имел вид и благополучно функционировал. Чтобы это сделать, необходимо в папке `templates` создать несколько файлов: `index.html` и `converter.html`. Первый файл отвечает за визуализацию дизайна и функционал калькулятора, а второй за конвертер соответственно. На рисунке 6 показан фрагмент кода, отвечающий за стиль определённой части сайта, в данном примере это «тело» сайта – его основная часть. На рисунке 7 представлен код, который отвечает за функционал кнопок нашего калькулятора. С помощью тега `<button>` создаётся кнопка и ей присваивается функция. На рисунке 8 изображён код, отвечающий за вывод результата в ходе проведения вычислений с некоторыми тригонометрическими функциями. Все значения возвращаются в радианах.

```

<style>
body {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  margin: 0;
  background-color: #7fb5ac;
  font-family: Arial, sans-serif;
}

```

Рисунок 6 – прописанный стиль для содержимого страницы

```

<div class="calculator">
  <b><div class="title">Онлайн-калькулятор для инженерных
расчётов</div></b>
  <div class="display" id="display">0</div>
  <div class="buttons">
    <button onclick="trig(&#39;sin&#39;)">sin</button>
    <button onclick="trig(&#39;asin&#39;)">asin</button>
    <button onclick="constant(&#39;PI&#39;)">π</button>
    <button onclick="memClear()">MC</button>
    <button onclick="memRecall()">MR</button>
    <button onclick="memSub()">M-</button>
    <button onclick="memAdd()">M+</button>

    <button onclick="trig(&#39;cos&#39;)">cos</button>
    <button onclick="trig(&#39;acos&#39;)">acos</button>
    <button onclick="constant(&#39;E&#39;)">e</button>
    <button onclick="addChar(&#39;(&#39;)">(</button>
    <button onclick="plusMinus()">+/-</button>
    <button onclick="addChar(&#39;)&#39;)">)</button>
    <button onclick="clearDisplay()">C</button>
  </div>
</div>

```

Рисунок 7 – ввод некоторых кнопок для калькулятора

```

switch(func) {
  case 'sin':
    result = Math.sin(num);
    break;
  case 'cos':
    result = Math.cos(num);
    break;
  case 'tan':
    result = Math.tan(num);
    break;
  case 'asin':
    result = Math.asin(num);
    break;
  case 'acos':
    result = Math.acos(num);
    break;
}

```

Рисунок 8 – кнопки тригонометрических функций, которые возвращают значения в радианах

Далее необходимо прописать остальные функции калькулятора (логарифм, возведение в квадрат, квадратный корень и т.д.). Рассмотрим реализацию функции памяти, чтобы калькулятор мог запоминать последние полученные значения, и проводить операции с ними. На рисунке 9 представлен код, реализующий память для калькулятора:

```
function memAdd() {
    memory += parseFloat(display.textContent);
}

function memSub() {
    memory -= parseFloat(display.textContent);
}

function memRecall() {
    display.textContent = memory.toString();
    newNumber = true;
}

function memClear() {
    memory = 0;
}
```

Рисунок 9 – функции памяти для калькулятора

Теперь нужно создать функцию, которая отвечает за выполнение математических операций в калькуляторе, обрабатывает случаи с ошибками и обновляет интерфейс пользователя в зависимости от результатов вычислений. Эта функция показана на рисунке 10:

```
function calculate() {
    try {
        let result;
        if (operator === 'power') {
            let secondOperand = parseFloat(display.textContent);
            result = Math.pow(firstOperand, secondOperand);
        } else {
            result = eval(display.textContent);
        }
        display.textContent =
parseFloat(result.toFixed(8)).toString();
        firstOperand = null;
        operator = null;
        newNumber = true;
    } catch (e) {
        display.textContent = 'Error';
        newNumber = true;
    }
}
```

Рисунок 10 – главная функция калькулятора

Для реализации конвертера необходимо создать выпадающие списки и для каждой единиц прописать коэффициенты. Список реализуется с помощью тегов `<select>` и `<option>`. На рисунке 11 показан код списков на примере длины:

```
<div class="converter-container">
  <div id="length" class="converter-section active">
    <div class="unit-group">
      <label>Из:</label>
      <input type="number" id="length-input"
value="1">
      <select id="length-from">
        <option value="m">Метры</option>
        <option value="km">Километры</option>
        <option value="cm">Сантиметры</option>
        <option value="mm">Миллиметры</option>
        <option value="mi">Мили</option>
        <option value="yd">Ярды</option>
        <option value="ft">Футы</option>
        <option value="in">Дюймы</option>
      </select>
    </div>
  </div>
```

Рисунок 11 – выпадающий список для единиц длины конвертера

Коэффициенты единиц измерения задаются постоянными переменными. Они необходимы для того, чтобы можно было переводить из одних единиц в другие, это реализуется с помощью произведения введённого значения на коэффициент, на который необходимо перевести. На рисунке 12 показаны коэффициенты на том же примере физической величины длины:

```
const conversions = {
  length: {
    m: 1,
    km: 1000,
    cm: 0.01,
    mm: 0.001,
    mi: 1609.344,
    yd: 0.9144,
    ft: 0.3048,
    in: 0.0254
  },
}
```

Рисунок 12 – общепринятые коэффициенты для единиц длины

После прописанных списков и коэффициентов для конвертера, необходимо прописать саму функцию конвертации. Для этого требуется принять входное значение и умножить его на коэффициент выбранной величины. Это показано на рисунке 13:

```
function convert(type, value, fromUnit, toUnit) {  
    const fromValue = value * conversions[type][fromUnit];  
    return fromValue / conversions[type][toUnit];  
}
```

Рисунок 13 – функция конвертации

Остаётся прописать заключительную часть кода конвертера – код предназначенный для создания функциональности конвертации различных единиц измерения (длина, скорость, масса, время, площадь и объем). Этот фрагмент кода представлен на рисунке 14:

```
['length', 'speed', 'mass', 'time', 'area', 'volume'].forEach(type => {  
    const input = document.getElementById(`${type}-input`);  
    const output = document.getElementById(`${type}-output`);  
    const fromSelect = document.getElementById(`${type}-from`);  
    const toSelect = document.getElementById(`${type}-to`);  
  
    [input, fromSelect, toSelect].forEach(element => {  
        element.addEventListener('input', () => {  
            const value = parseFloat(input.value);  
            if (!isNaN(value)) {  
                const result = convert(type, value, fromSelect.value,  
toSelect.value);  
                output.value = result.toFixed(6);  
            } else {  
                output.value = '';  
            }  
        });  
    });  
});
```

Рисунок 14 – обработка изменений во всех конвертерах

Четвёртый этап:

Последним этапом является протестировать сайт, то есть проверить его работоспособность, функциональность, правильность выполнения всех команд и правильное отображение всех кнопок.

Функциональное и оценочное тестирование

После успешного написания кода, необходимо провести тестирование. Оно будет подразделяться на функциональное и оценочное.

Функциональное тестирование проверяет работоспособность калькулятора и конвертера, то есть правильность работы кнопок, отвечающих за ввод и вывод значений, кнопок арифметических операций и т.д.

Оценочное тестирование, показывает обратную связь нескольких пользователей об удобстве пользования и удобстве эксплуатации сайтом. Данные о функциональном и оценочном тестировании показаны на таблицах 1 - 3:

| № теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
|---------|---|---|---|---|---------------------------|
| 1 | Проверка корректности работы кнопки перехода между страницами | Нажатие на кнопку перехода между страницами | Ожидается переход на другую страницу | Переход на другую страницу | Успешная работа программы |
| 2 | Проверка корректности работы кнопок для ввода цифр | Нажатие на кнопку с цифрой | Ожидается вывод цифры на экран калькулятора | Вывод цифры на экран калькулятора | Успешная работа программы |
| 3 | Проверка корректности работы кнопок арифметических операций и кнопки «=» на | Нажатие кнопок: ввода цифры, «-», ввода вычитаемой цифры, «=» | Ожидается вывод каждого действия на экран калькулятора, и после нажатия | Вывод действий на экран, и после нажатия «=» вывод результата | Успешная работа программы |

| | | | | | |
|---|---|---|---|--|---------------------------------|
| | примере вычитания | | «=» вывод результата | | |
| 4 | Проверка корректности работы кнопок тригонометричес ких функций на примере «atg» | Нажатие кнопки ввода цифры, кнопки «atg» и кнопки «=» | Ожидается вывод каждого действия на экран калькулятора, и после нажатия «=» вывод результата | Вывод действий на экран, и после нажатия «=» вывод результата | Успешная работа программы |

Таблица 1 – результаты функционального тестирования калькулятора

| № теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
|------------|---|--|--|---|---------------------------------|
| 1 | Проверка корректности работы кнопок переключения между величинами | Нажатие на кнопку переключения между величинами | Ожидается переключение между величинами | Переключение между величинами | Успешная работа программы |
| 2 | Проверка корректности работы кнопки выпадающего списка и выбора единицы величины | Нажатие на кнопку выпадающего списка и выбора единицы величины | Ожидается появление выпадающего списка и переключение на другую единицу величины | Появление выпадающего списка и переключение на другую единицу величины | Успешная работа программы |

| | | | | | |
|---|---|--|--|--|---------------------------------|
| 3 | Проверка корректности работы ввода численного значения и перевода в другие единицы | Ввод численного значения в поле для исходных единиц | Ожидается ввод численного значения и отображение его на поле для исходных единиц | Ввод численного значения и отображение его на поле для исходных единиц | Успешная работа программы |
|---|---|--|--|--|---------------------------------|

Таблица 2 – результаты функционального тестирования конвертера

В таблицах 1 и 2 были представлены результаты функционального тестирования. В таблице 3 будут представлены результаты оценочного тестирования, в котором приняли участие 5 пользователей. Они поставили оценки за удобство использования и удобство эксплуатации. В последней строке рассчитана средняя оценка за эти два критерия:

| № пользователя | Удобство использования | Удобство эксплуатации |
|-------------------|------------------------|-----------------------|
| 1 | 9 | 8 |
| 2 | 9 | 9 |
| 3 | 10 | 7 |
| 4 | 7 | 8 |
| 5 | 9 | 9 |
| Средняя оценка | 8,8 | 8,2 |

Таблица 3 – результаты оценочного тестирования

Заключение

В итоге был создан сайт с рабочими инженерным калькулятором и конвертером физических величин, который поможет проводить арифметические и инженерные расчёты, а также переводить из одних физических единиц в другие. Была продемонстрирована работа с дизайном сайта и его кодом. Можно сделать вывод о том, что была проделана трудоёмкая и время затратная работа по написанию кода и реализации большого количества функций калькулятора, а также конвертера. Подводя к итогам, можно сказать, что задачи проекта выполнены, а цель проекта достигнута.

Список литературы

1. Баранов, С. В. "Основы работы с Django." М.: Издательство, 2021.
2. Кириченко, А.А. "Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков." М.: Наука и техника, 2021.
3. Силин, П. А. "Проектирование и разработка веб-приложений." М.: Издательство, 2020.
4. Степанов, И. А. "Методы тестирования программного обеспечения." М.: Издательство, 2017.
5. Шабанов, Д. Ю. "Дизайн пользовательского интерфейса." М.: Издательство, 2020.
6. Шилдт, Г. "Java. Полное руководство" М.: Диалектика-Вильямс, 2018.