Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут  ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Практична робота № 3

з курсу: «Основи Веб-програмування»

**Виконав:**  
студент 2-го курсу,  
групи ТВ-33  
Козінченко Тимофій Олександрович

 Посилання на GitHub репозиторій: <https://github.com/efamir/PW03TV-33_Kozinchenko_Tymofii_Oleksandrovych>

**Перевірив:**

Недашківський О.Л.

Київ 2025

**Завдання 1.**

Створіть веб калькулятор розрахунку прибутку від сонячних електростанцій з

встановленою системою прогнозування сонячної потужності

**Виконання**

Перед виконанням завдання потрібно було знайти спосіб розраховувати значення частки енергіїї, що генерується без небалансів. Формула на рисунку 1.1:

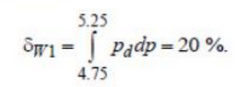


Рисунок 1.1 – формула частки енергії

Де pd – це формула нормального розподілу потужності (див рисунок. 1.2):

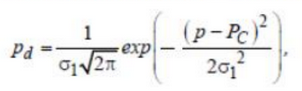
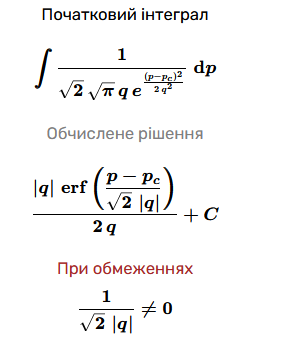


Рисунок 1.2 – формула нормального розподілу потужності

Розраховувати інтеграл у JavaScript не так просто, тому було вирішено завчасно проінтегрувати форму та використовувати її у обчисленнях. Для точного розрахунку було використано калькулятор:



Тепер ми можемо використовувати отриману формулу у розрахунках. Проте, у стандартній бібліотеці JavaScript нема формули для розрахунку erf, тому її імплементація була зроблена завдяки аппроксимації функції помилки за Абрамовицем і Стегином.

Програму було написано використовуючи HTML для розмітки вебсторінки та JavaScript для підрахунку результатів та їх відображення в HTML документі. Для отримання вхідних даних використовувались теги «input» з атрибутами:

* «type=”number”» для отримання тільки числових значень
* «id=”<<id\_here>>”» для ідентифікації кожного тегу

Також для кожного з них було також додано тег label форматом «<label for="<<input\_id\_here>>"><<label text>></label>». Для тригеру функції підрахунку було додано кнопку яка при натисканні визиває функцію в JavaScript файлі (про нього пізніше).

HTML файл ще містить у собі теги «<span id="<<id>>"></span>» для подальшою зміни в них тексту («innerText») JavaScript скриптом для відображення результатів.

Код HTML файлу можна побачити на рисунку 1.3.

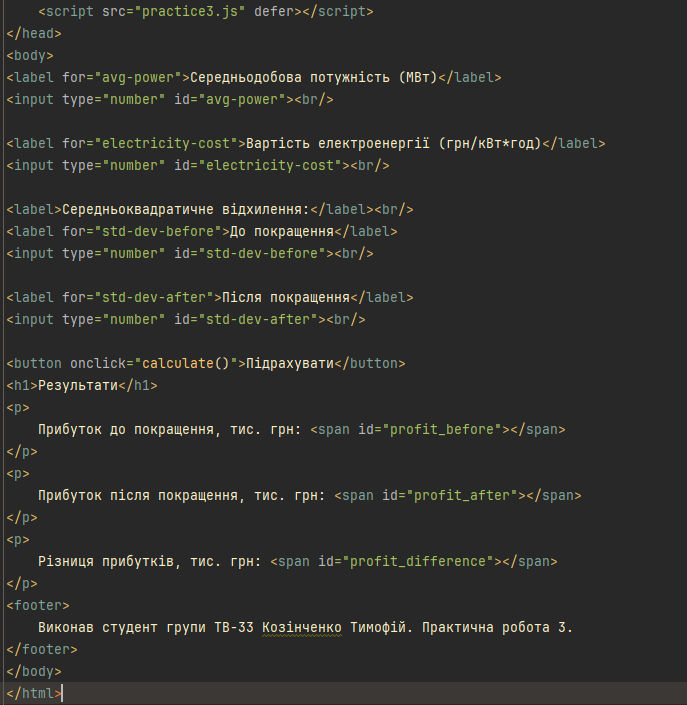


Рисунок 1.3 – Код для вводу та виводу результатів обрахунків

Залишилось лише створити скрипт який би за викликом функції «calculate», виводив би результати в HTML документ. Для взаємодії з HTML документом використовувався метод «document.getElementById(<<id\_here>>)». Після чого, вже за полями «value» «innerText» отримувались значення вводу та виводились результати відповідно. Код скрипта зображено на рисунку 1.4, 1.5 та 1.6.

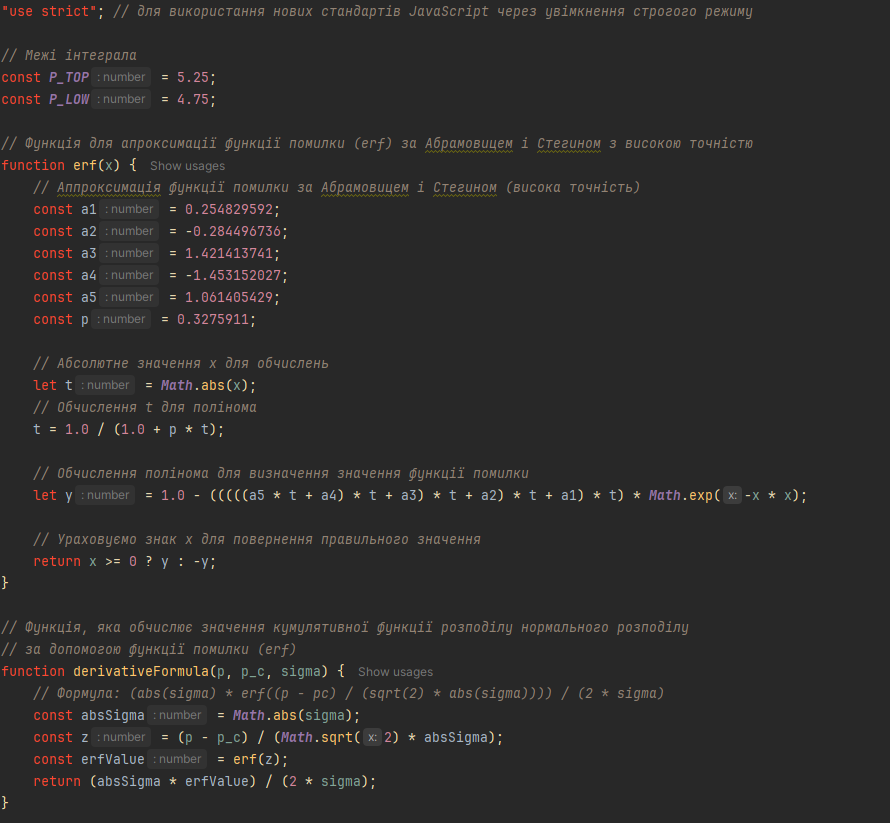


Рисунок 1.4 – код JavaScript скрипту. Частина 1.

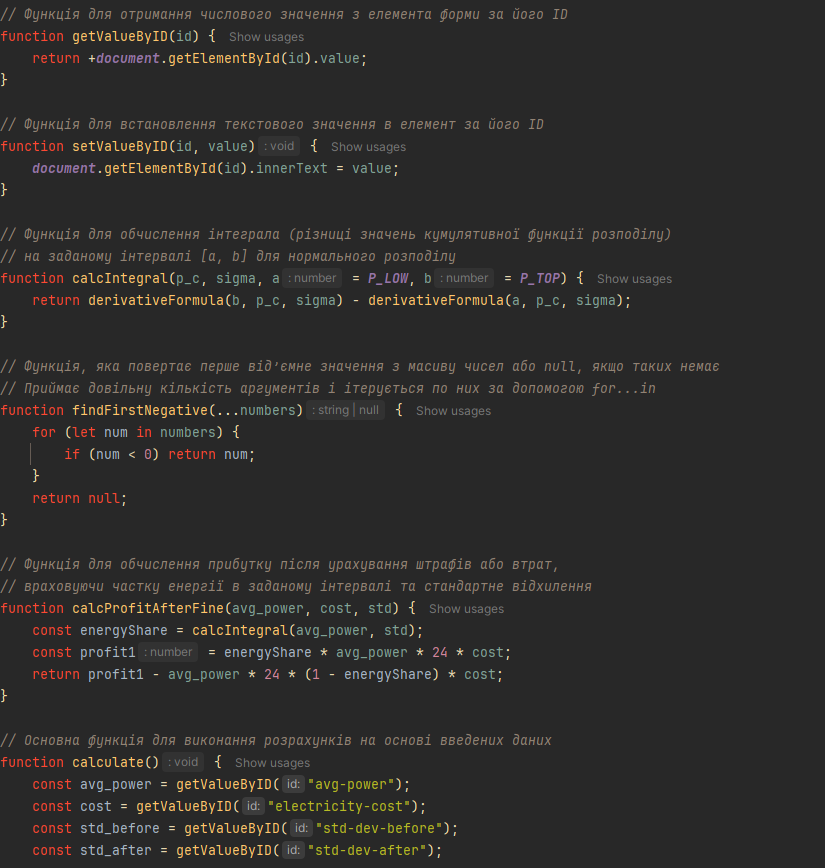


Рисунок 1.5 – Код JavaScript скрипту. Частина 2

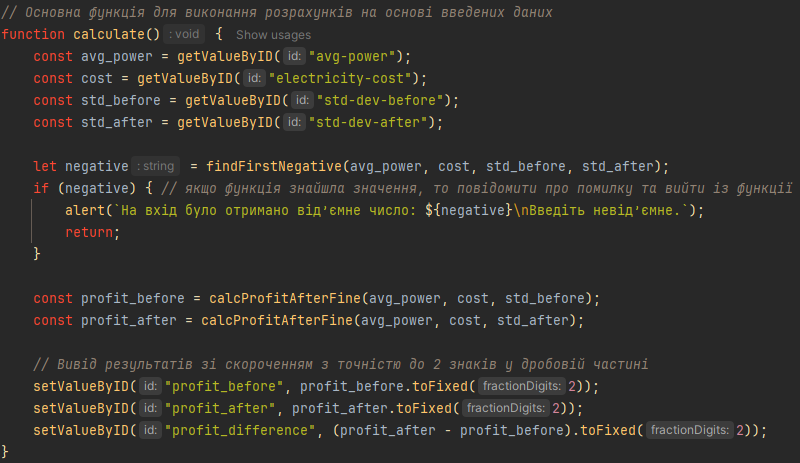


Рисунок 1.6 – Код JavaScript скрипту. Частина 3

Скрипт було завантажено у HTML документ завдяку тегу «<script src="practice3.js" defer></script>». «defer» в ньому означає, що скрипт чекає поки не завантажиться весь документ.

Наприкінець, було вирішено покращити інтерфейс калькулятора, тому було написано css код. У CSS я використав базові властивості, такі як font-family, щоб установити шрифт Arial для читабельності, і max-width із margin: auto, щоб сторінка була центрована та зручна для перегляду. Додав фонові кольори типу background-color: #f5f5f5 для світлого фону й color: #333 для тексту, щоб забезпечити гарний контраст. Використовував border-radius для закруглених кутів у кнопок і полів вводу, а також hover для кнопки, щоб додати інтерактивність при наведенні курсору. Усе це зроблено просто, але стильно, щоб сторінка виглядала привабливо, але не перевантаженою, як ми вчилися на заняттях із CSS.

**Перевірка виконання на контрольному прикладі**

На рисунку 1.7 зображено виконання вебзастосунку на котрольному прикладі

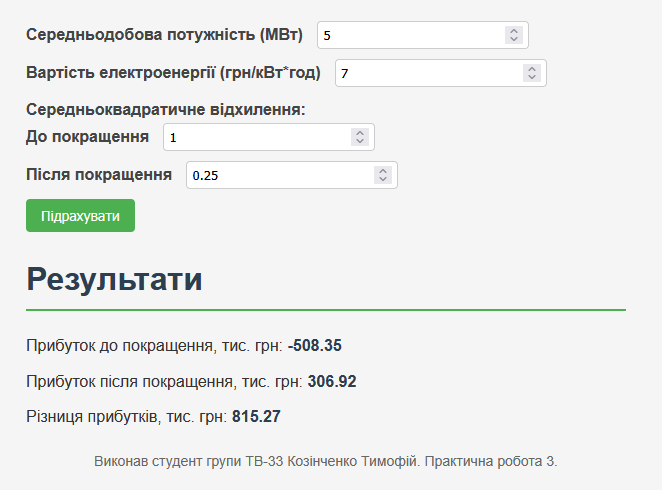


Рисунок 1.7 – Результат виконання застосунку на контрольному прикладі

Бачимо, що значення співпали, але іноді є невеличка похибка (в дробах). Причиною є скорочення чисел у контрольному прикладі.

**Висновки**

Під час виконання практичної роботи №3 я повторив як взаємодіяти з документом вебсторінки використовуючи мову програмування JavaScript: обробляти натискання на кнопку, зчитувати вміст полів та оновлювати контент на сторінці. Оновив навички написання коду з арифметичними операціями та використання функцій на мові програмування JavaScript. На практиці використав CSS.

Також було пригадано як можна обчислювати інтеграли для подальшого використання в обчисленнях.