Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 5

Варіант 5

з курсу: «Програмування вебзастосунків»

**Виконала:**  
студентка 4-го курсу,  
групи ТВ-11

Барабаш Маріна Володимирівна

Посилання на GitHub репозиторій:

https://github.com/Aylosteraa/PW\_TV-11-Barabash\_Marina\_Volodymyrivna.git

**Перевірив:**

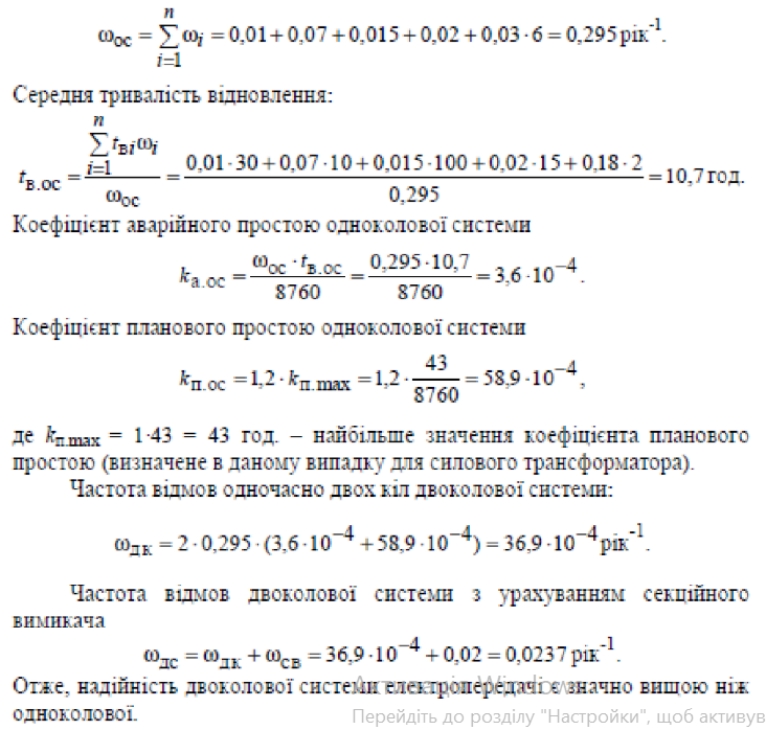
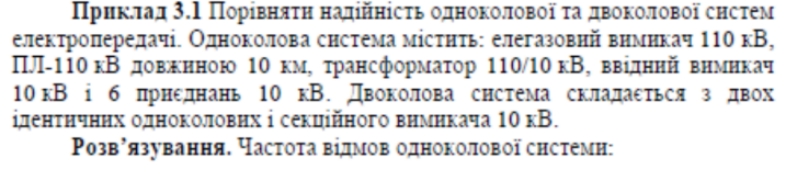
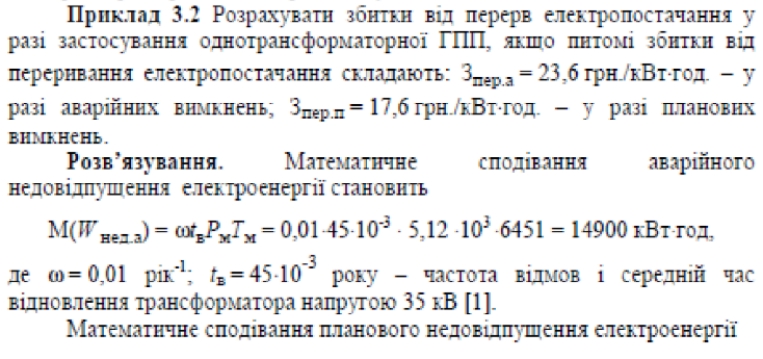
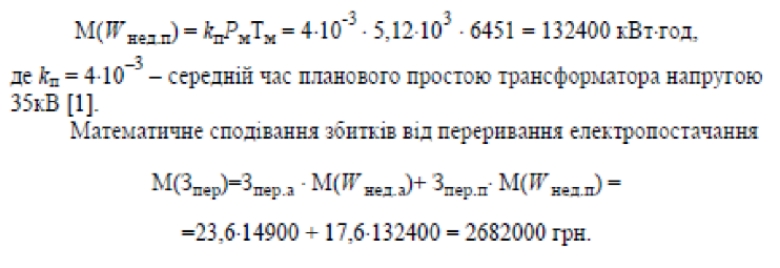
Недашківський О.Л.

Київ 2025

**Лабораторна робота № 5**

**Завдання:**

Створіть "Веб калькулятор для порівняння надійності одноколової та двоколової систем електропередачі та розрахунку збитків від перерв електропостачання у разі застосування однотрансформаторної ГТП."

**Хід роботи:**

package main

import (

    "fmt"

    "html/template"

    "log"

    "net/http"

    "strconv"

)

func roundFloat(value float64, precision int) string {

    format := fmt.Sprintf("%%.%df", precision)

    return fmt.Sprintf(format, value)

}

type ReliabilityParameters struct {

    omega float64

    tV    int

    mu    float64

    tP    int

}

var data = map[string]ReliabilityParameters{

    "PL-110 kV":                    {0.007, 10, 0.167, 35},

    "PL-35 kV":                     {0.02, 8, 0.167, 35},

    "PL-10 kV":                     {0.02, 10, 0.167, 35},

    "CL-10 kV (Trench)":            {0.03, 44, 1.0, 9},

    "CL-10 kV (Cable Channel)":     {0.005, 18, 1.0, 9},

    "T-110 kV":                     {0.015, 100, 1.0, 43},

    "T-35 kV":                      {0.02, 80, 1.0, 28},

    "T-10 kV (Cable Network)":      {0.005, 60, 0.5, 10},

    "T-10 kV (Overhead Network)":   {0.05, 60, 0.5, 10},

    "B-110 kV (Gas-Insulated)":     {0.01, 30, 0.1, 30},

    "B-10 kV (Oil)":                {0.02, 15, 0.33, 15},

    "B-10 kV (Vacuum)":             {0.05, 15, 0.33, 15},

    "Busbars 10 kV per Connection": {0.03, 2, 0.33, 15},

    "AV-0.38 kV":                   {0.05, 20, 1.0, 15},

    "ED 6,10 kV":                   {0.1, 50, 0.5, 0},

    "ED 0.38 kV":                   {0.1, 50, 0.5, 0},

}

func CalculatorHandler1(w http.ResponseWriter, r \*http.Request) {

    if r.Method == http.MethodPost {

        // Отримуємо значення з форми

        wOc := 0.0

        tVOc := 0.0

        for key := range data {

            amount, \_ := strconv.Atoi(r.FormValue(key))

            indicator := data[key]

            if amount > 0 {

                wOc += float64(amount) \* indicator.omega

                tVOc += float64(amount) \* indicator.omega \* float64(indicator.tV)

            }

        }

        tVOc /= wOc

        kAOc := (tVOc \* wOc) / 8760

        kPOs := 1.2 \* 43 / 8760

        wDk := 2 \* wOc \* (kAOc + kPOs)

        wDc := wDk + 0.02

        precision := 5

        tmpl, \_ := template.ParseFiles("templates/calculator1.html")

        tmpl.Execute(w, map[string]string{

            "wOc":  roundFloat(wOc, precision),

            "tVOc": roundFloat(tVOc, precision),

            "kAOc": roundFloat(kAOc, precision),

            "kPOs": roundFloat(kPOs, precision),

            "wDk":  roundFloat(wDk, precision),

            "wDc":  roundFloat(wDc, precision),

        })

        return

    }

    tmpl, \_ := template.ParseFiles("templates/calculator1.html")

    tmpl.Execute(w, nil)

}

func CalculatorHandler2(w http.ResponseWriter, r \*http.Request) {

    if r.Method == http.MethodPost {

        zPerA, \_ := strconv.ParseFloat(r.FormValue("zPerA"), 64)

        zPerP, \_ := strconv.ParseFloat(r.FormValue("zPerP"), 64)

        omega, \_ := strconv.ParseFloat(r.FormValue("omega"), 64)

        tV, \_ := strconv.ParseFloat(r.FormValue("tV"), 64)

        Pm, \_ := strconv.ParseFloat(r.FormValue("Pm"), 64)

        Tm, \_ := strconv.ParseFloat(r.FormValue("Tm"), 64)

        kP, \_ := strconv.ParseFloat(r.FormValue("kP"), 64)

        mWnedA := omega \* tV \* Pm \* Tm

        mWnedP := kP \* Pm \* Tm

        mZper := zPerA\*mWnedA + zPerP\*mWnedP

        precision := 2

        tmpl, \_ := template.ParseFiles("templates/calculator2.html")

        tmpl.Execute(w, map[string]string{

            "mWnedA": roundFloat(mWnedA, precision),

            "mWnedP": roundFloat(mWnedP, precision),

            "mZper":  roundFloat(mZper, precision),

        })

        return

    }

    tmpl, \_ := template.ParseFiles("templates/calculator2.html")

    tmpl.Execute(w, nil)

}

func homePage(w http.ResponseWriter, r \*http.Request) {

    tmpl, err := template.ParseFiles("templates/index.html")

    if err != nil {

        http.Error(w, "Помилка завантаження сторінки", http.StatusInternalServerError)

        return

    }

    tmpl.Execute(w, nil)

}

func main() {

    http.Handle("/static/", http.StripPrefix("/static/", http.FileServer(http.Dir("static"))))

    http.HandleFunc("/", homePage)

    http.HandleFunc("/calculator1", CalculatorHandler1)

    http.HandleFunc("/calculator2", CalculatorHandler2)

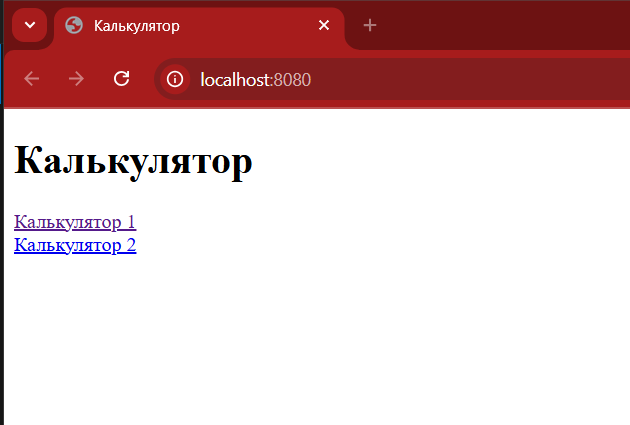
    port := ":8080"

    fmt.Println("Server running on http://localhost" + port)

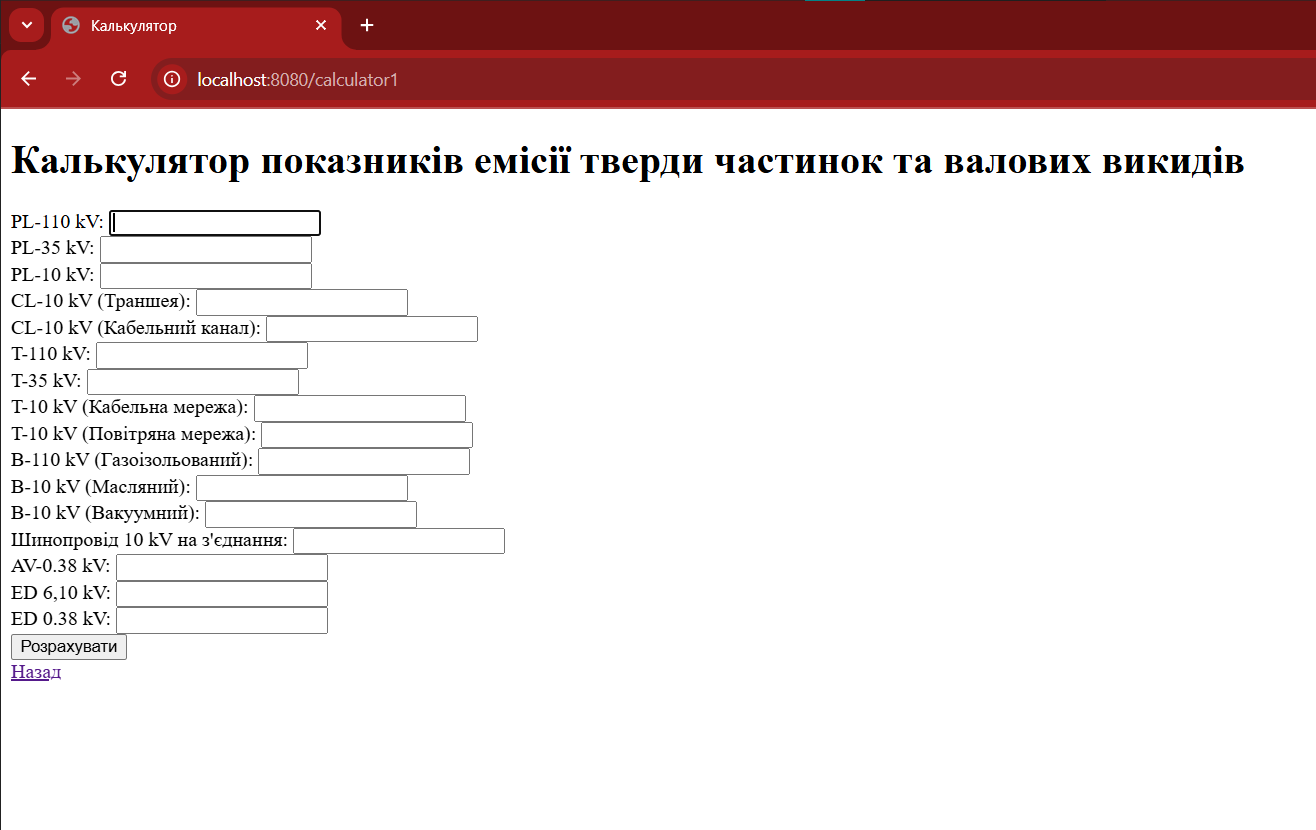
    log.Fatal(http.ListenAndServe(port, nil))

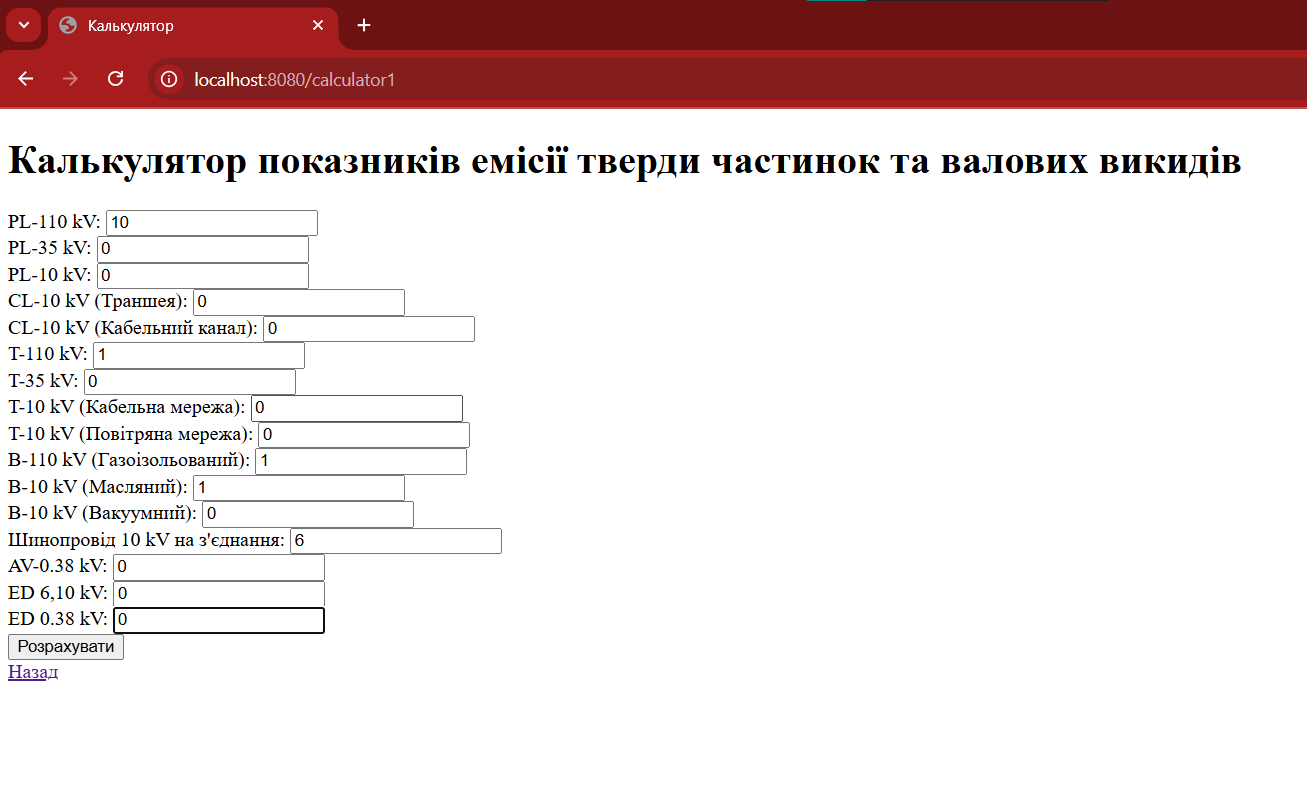
}

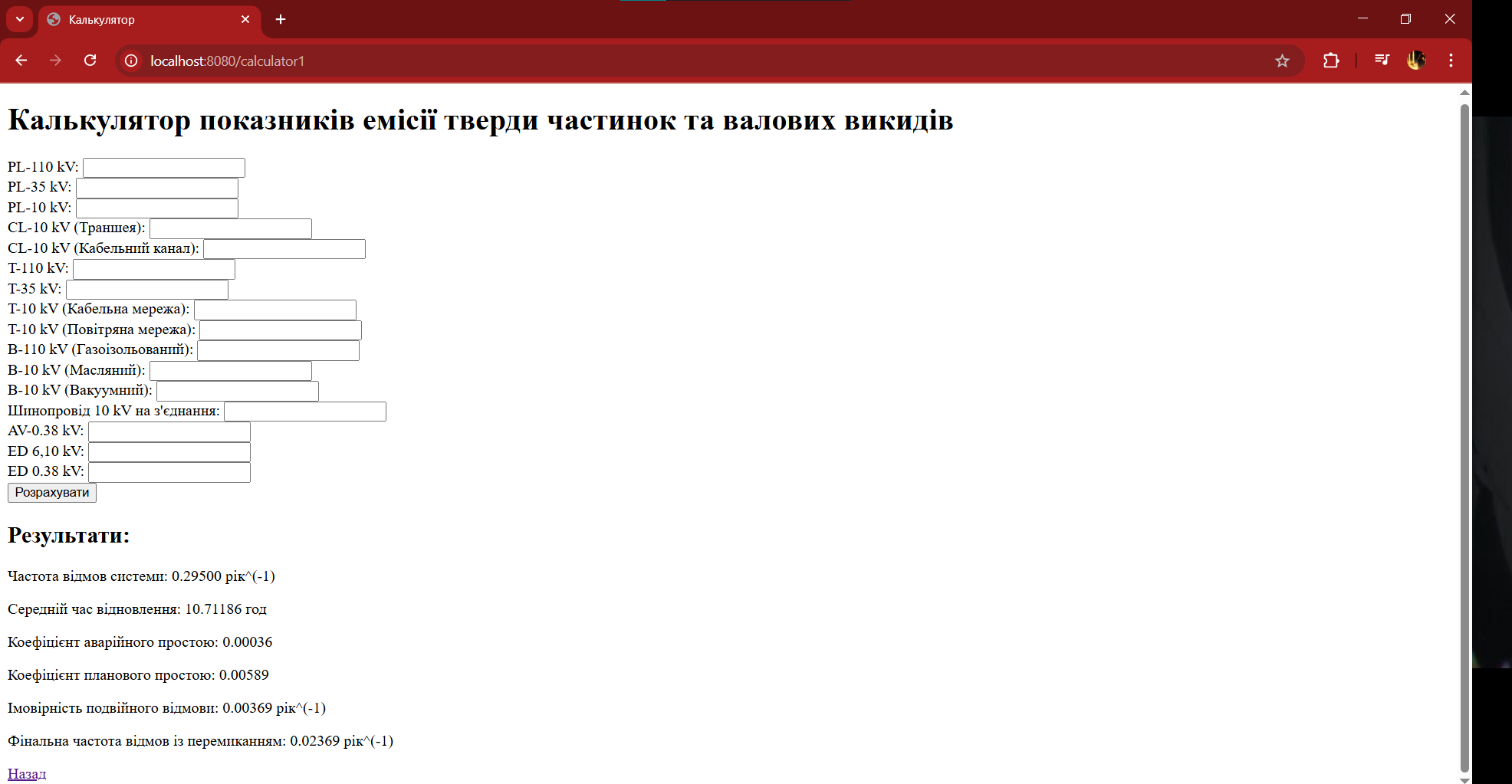
Головна сторінка



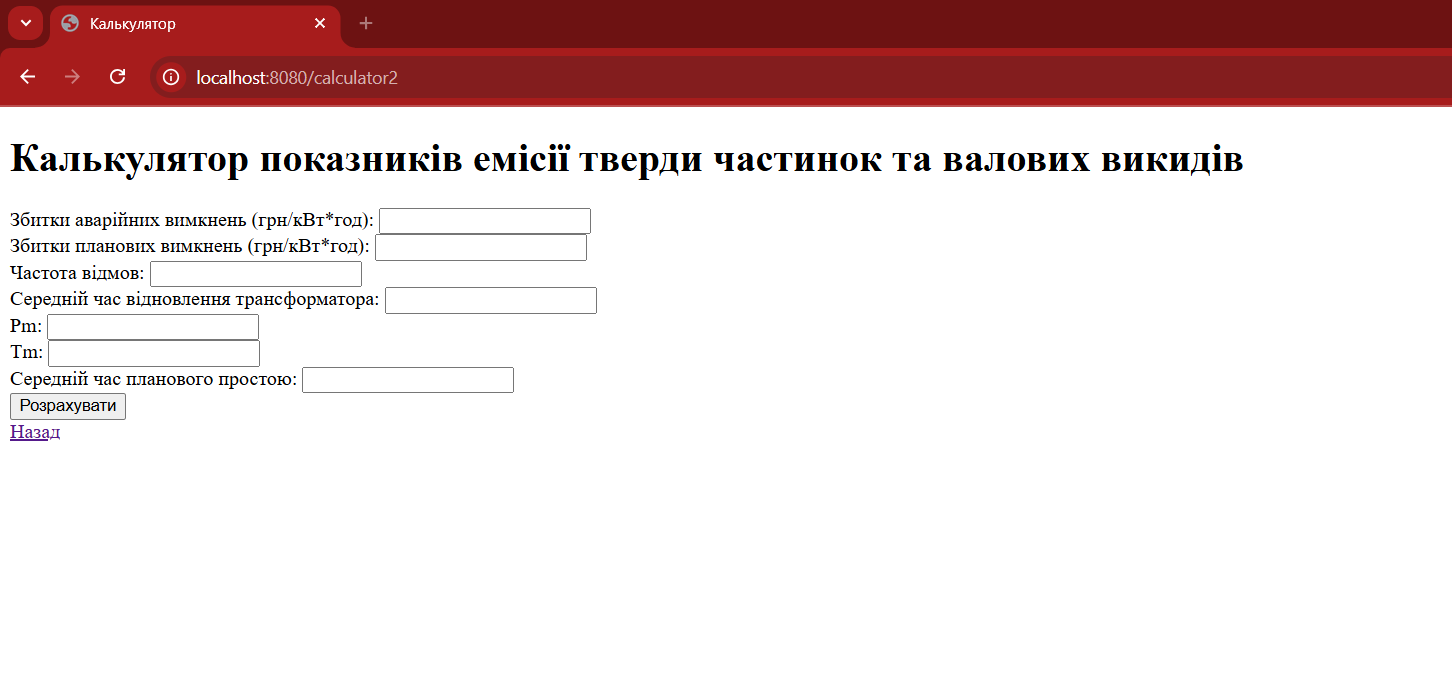
Калькулятор показників емісії твердих частинок та валових викидів

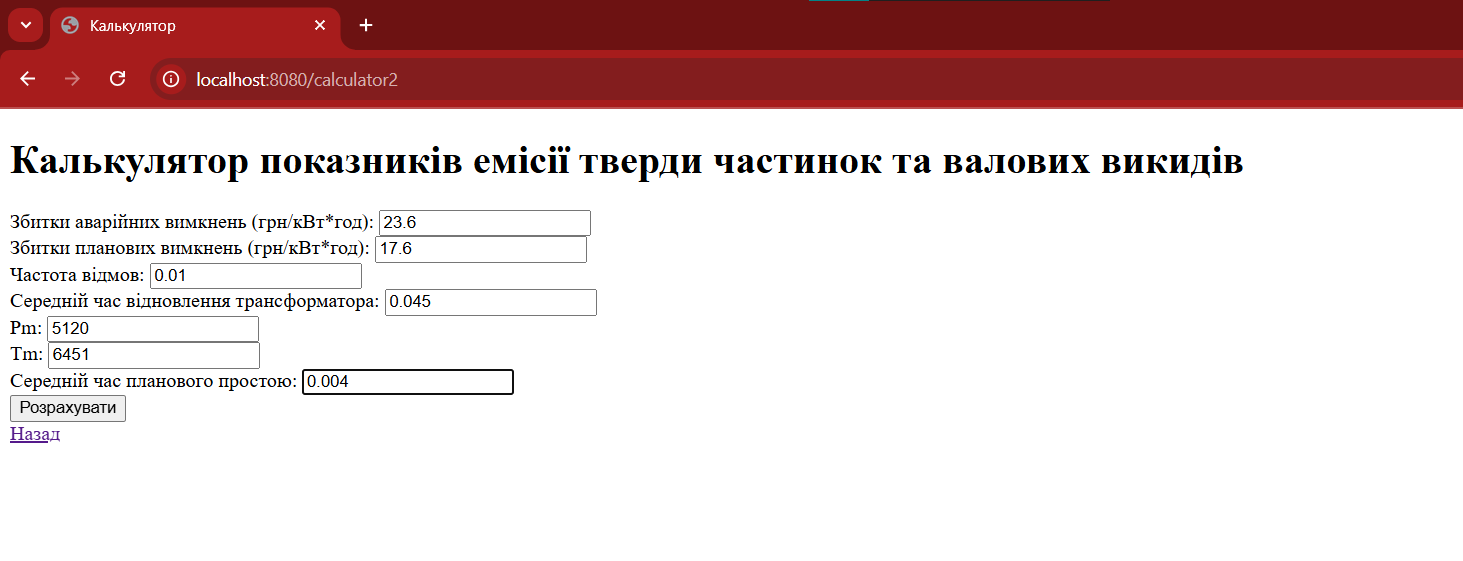


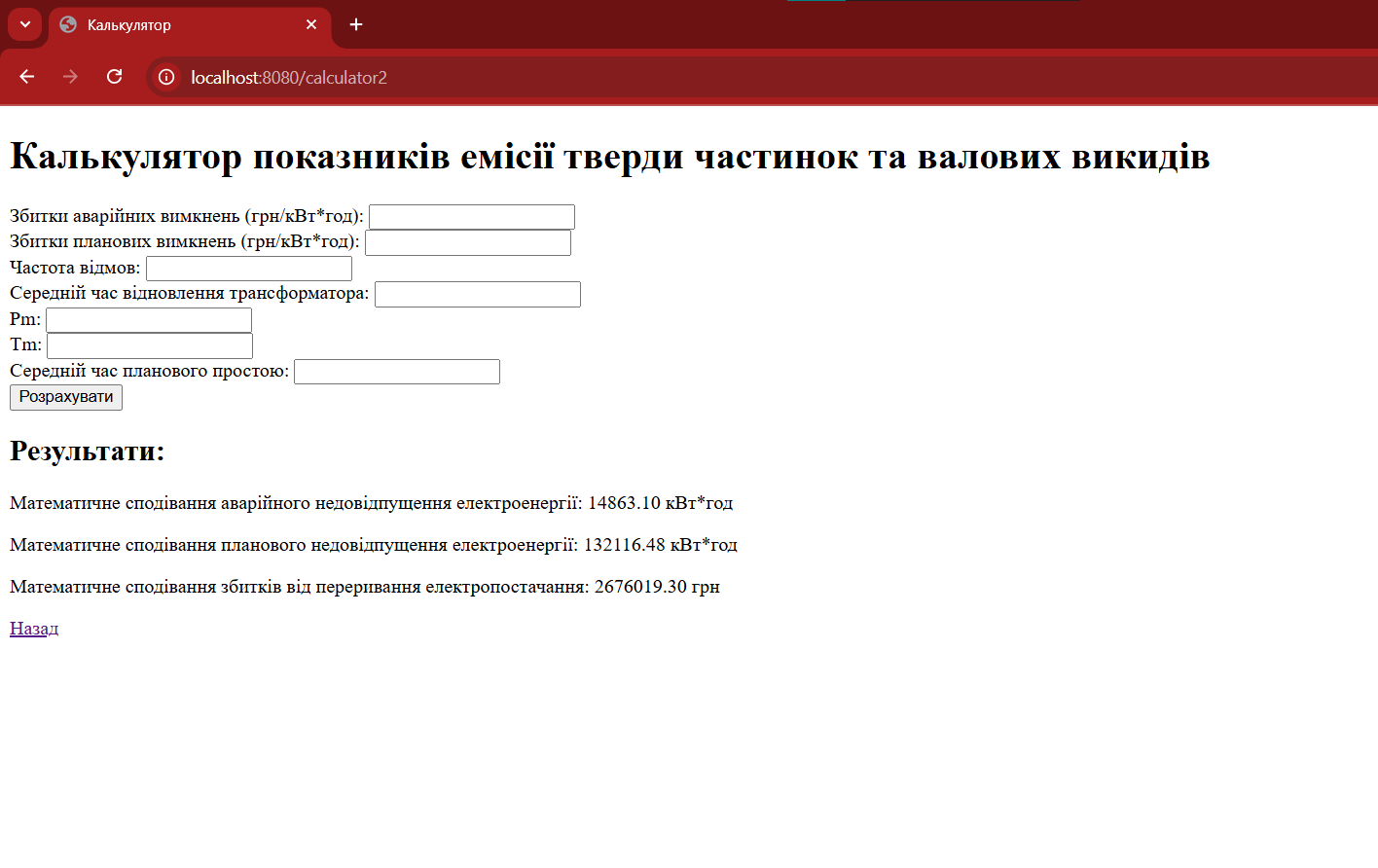




**Калькулятор показників емісіх твердих частинок та валових викидів**







**Висновок:**

В результаті виконання лабораторної роботи №5 було створено калькулятори мовою програмування Go для порівняння надійності одноколової та двоколової систем електропередачі та розрахунку збитків від перерв електропостачання у разі застосування однотрансформаторної ГТП.

.