Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 1

Варіант 5

з курсу: «Веб калькулятор для розрахунку

складу сухої та горючої маси палива та нижчої теплоти згоряння

для робочої, сухої та горючої маси за заданим складом компонентів

палива.»

**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:**

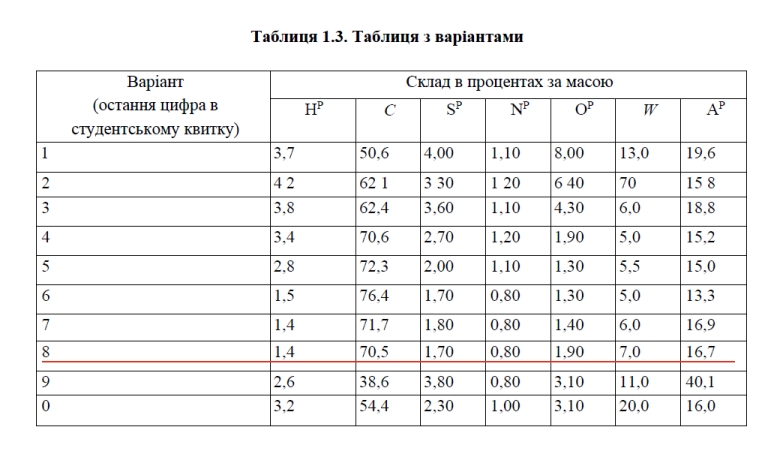
Недашківський О.Л.

Київ 2024/2025

Лабораторна робота №1

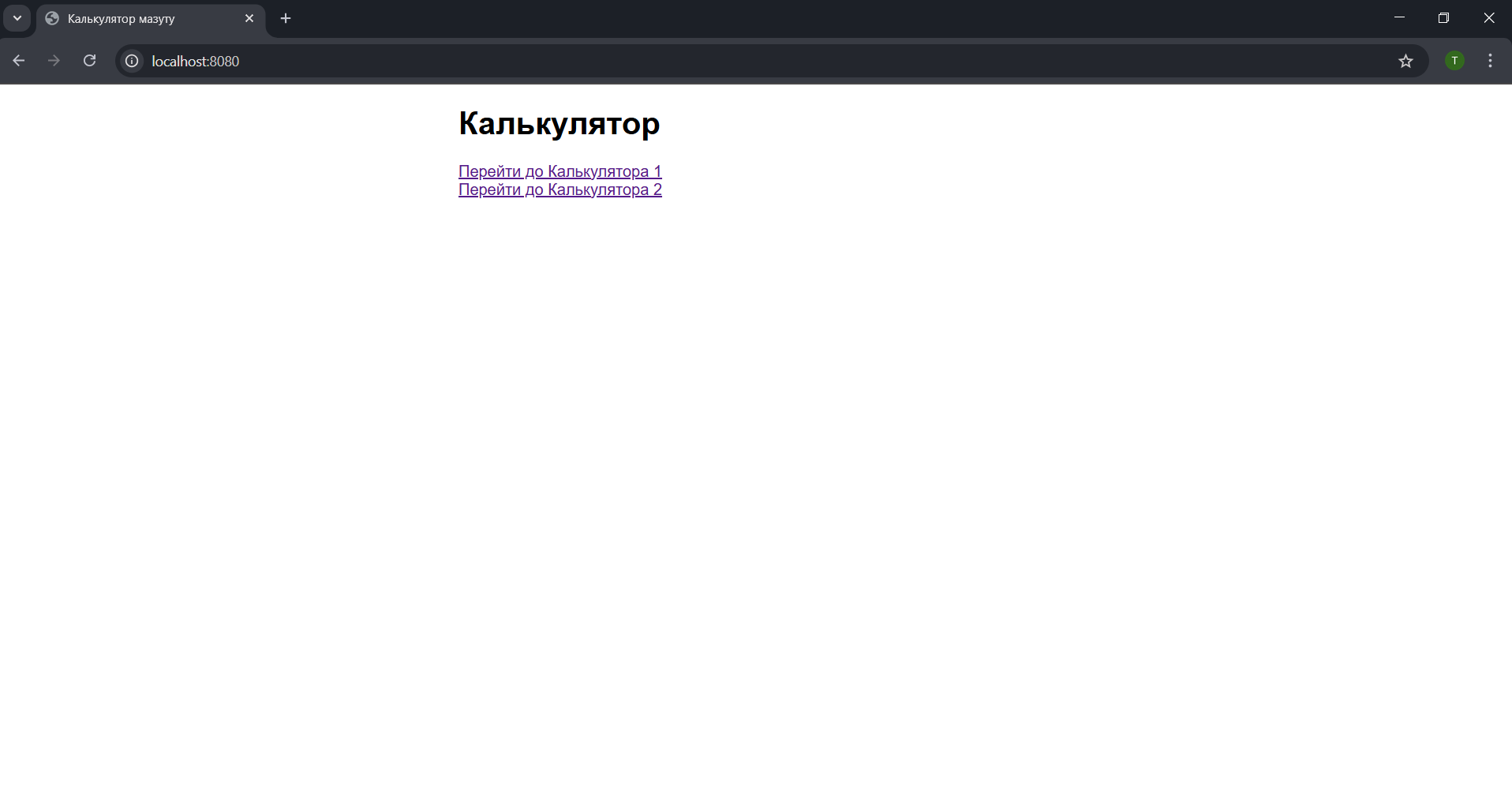
**Завдання:**

Написати веб калькулятор для розрахунку складу сухої та горючої маси палива та нижчої теплоти згоряння для робочої, сухої та горючої маси за заданим складом компонентів палива, що задаються у вигляді значень окремих компонентів типу: HP, %; CP, %; SP, %; NP, %; OP, %; WP, %; AP, % (див. табл. 1.3.).



**Хід виконання:**

Реалізуємо дизайн сторінки, на якій будуть наявні поля, де будуть здійснюватись переходи між двома калькуляторами.



**Код програми:**

package main

import (

    "fmt"

    "log"

    "net/http"

    "text/template"

    "lab1/calculators"

)

// Обробник для головної сторінки

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"))))

    // Налаштування маршрутів

    http.HandleFunc("/", homePage)

    http.HandleFunc("/calculator1", calculators.CalculatorHandler1)

    http.HandleFunc("/calculator2", calculators.CalculatorHandler2)

    // Запуск сервера

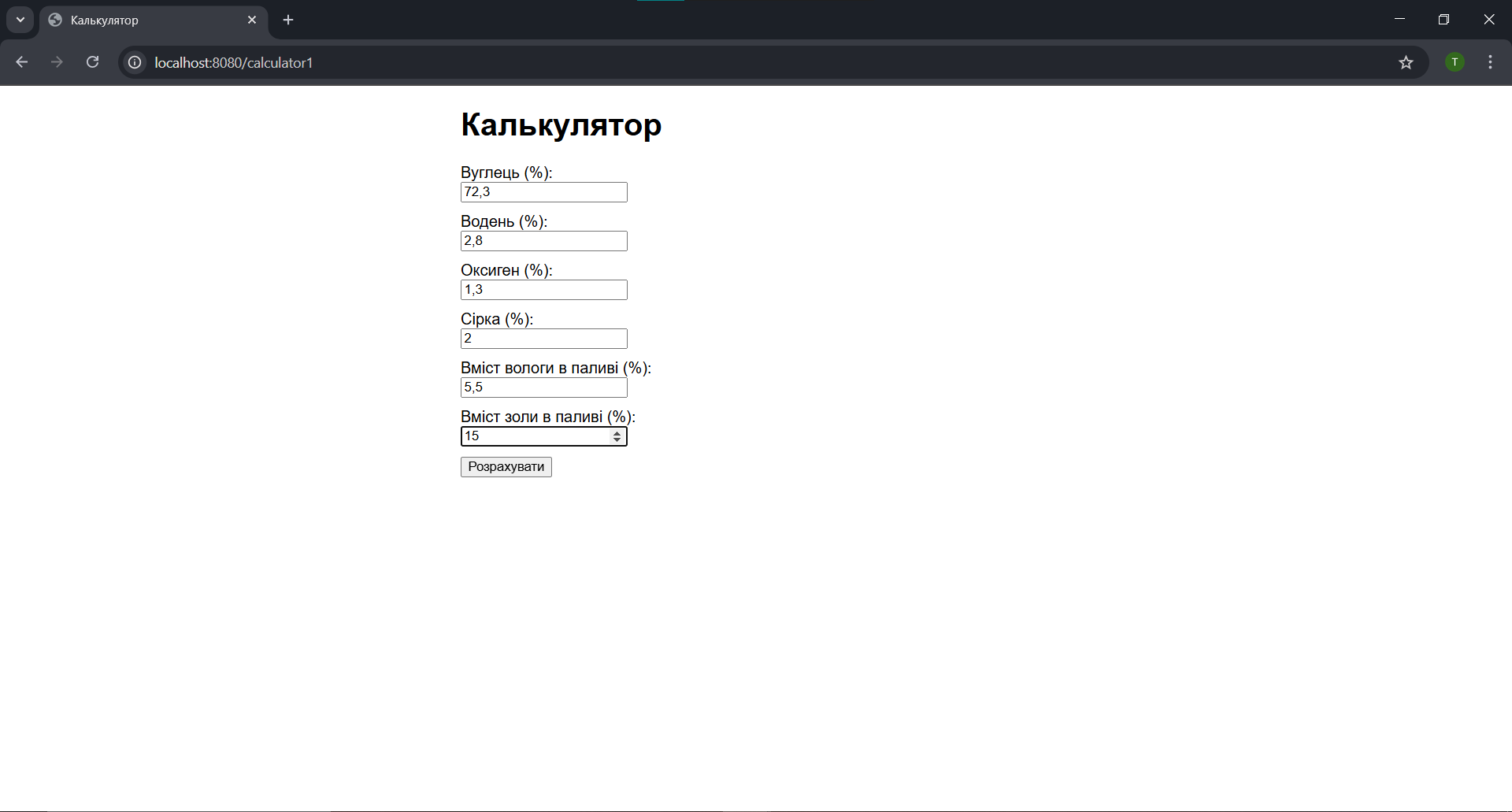
    port := ":8080"

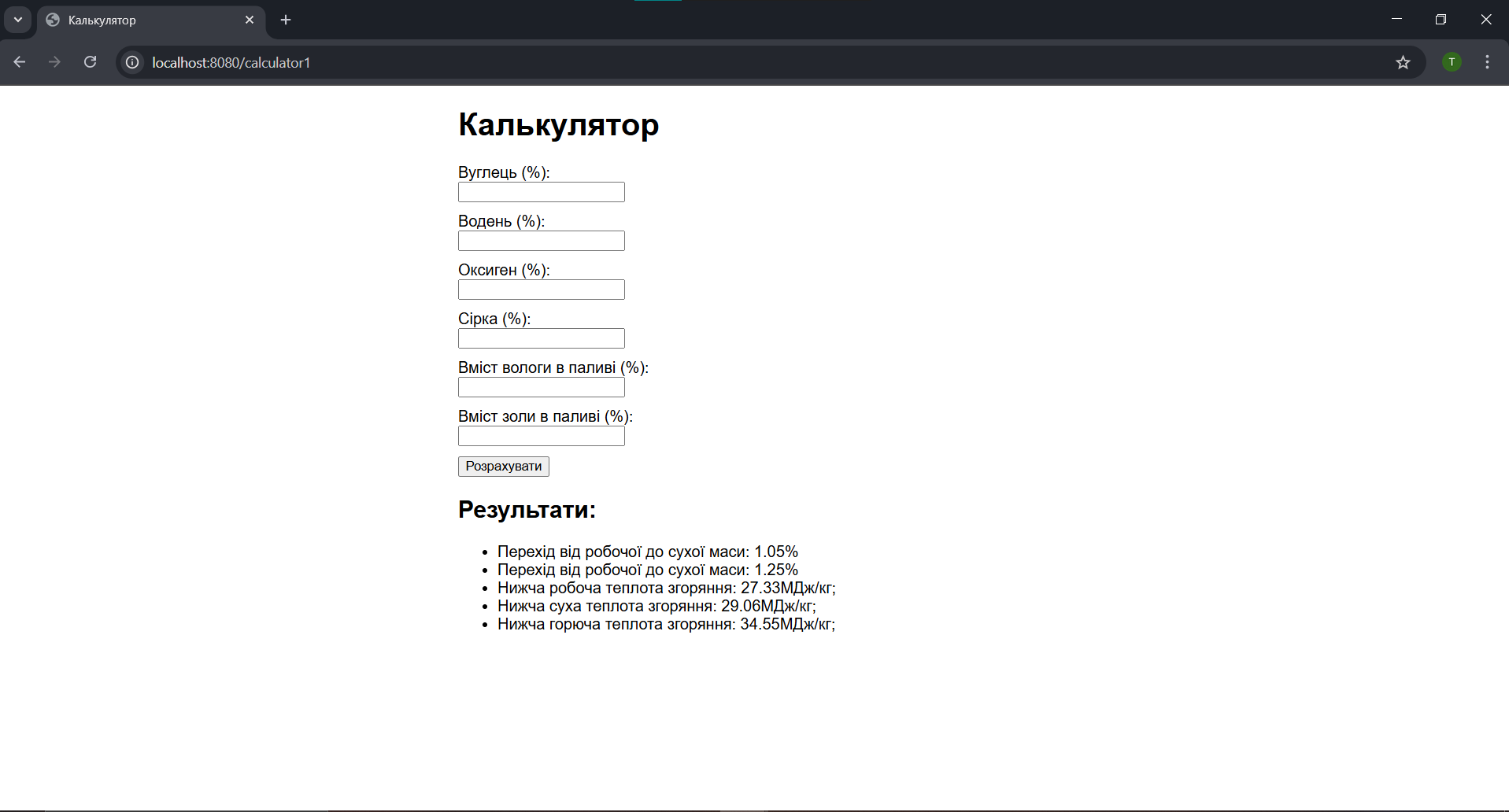
    fmt.Println("Server running on http://localhost" + port)

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

}

Тепер сторінку, де вводитися, потрібні нам коефіцієнти, та виводитися фінальні результати наших розрахунків для першого калькулятора:





**Код програми для розрахунків:**

package calculators

import (

    "html/template"

    "net/http"

)

// Структура для збереження даних форми та результатів

type Calculation1 struct {

    Carbon              float64

    Hydrogen            float64

    Oxygen              float64

    Sulfur              float64

    Moisture            float64

    Ash                 float64

    CoefficientWtoD     float64

    CoefficientWtoC     float64

    HeatWorkingMass     float64

    HeatDryMass         float64

    HeatCombustibleMass float64

    Calculated          bool

}

// Сторінка

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

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

    if err != nil {

        http.Error(w, "Error loading template", http.StatusInternalServerError)

        return

    }

    // Ініціалізація структури

    data := Calculation1{}

    if r.Method == http.MethodPost {

        // Отримання даних з форми

        data.Carbon = parseInput(r.FormValue("carbon"))

        data.Hydrogen = parseInput(r.FormValue("hydrogen"))

        data.Oxygen = parseInput(r.FormValue("oxygen"))

        data.Sulfur = parseInput(r.FormValue("sulfur"))

        data.Ash = parseInput(r.FormValue("ash"))

        data.Moisture = parseInput(r.FormValue("moisture"))

        // Розрахунки

        data.CoefficientWtoD = formatValue(100 / (100 - data.Moisture))

        data.CoefficientWtoC = formatValue(100 / (100 - data.Moisture - data.Ash))

        data.HeatWorkingMass = formatValue((339\*data.Carbon + 1030\*data.Hydrogen - 108.8\*(data.Oxygen-data.Sulfur) - 25\*data.Moisture) / 1000)

        data.HeatDryMass = formatValue((data.HeatWorkingMass + 0.025\*data.Moisture) \* 100 / (100 - data.Moisture))

        data.HeatCombustibleMass = formatValue((data.HeatWorkingMass + 0.025\*data.Moisture) \* 100 / (100 - data.Moisture - data.Ash))

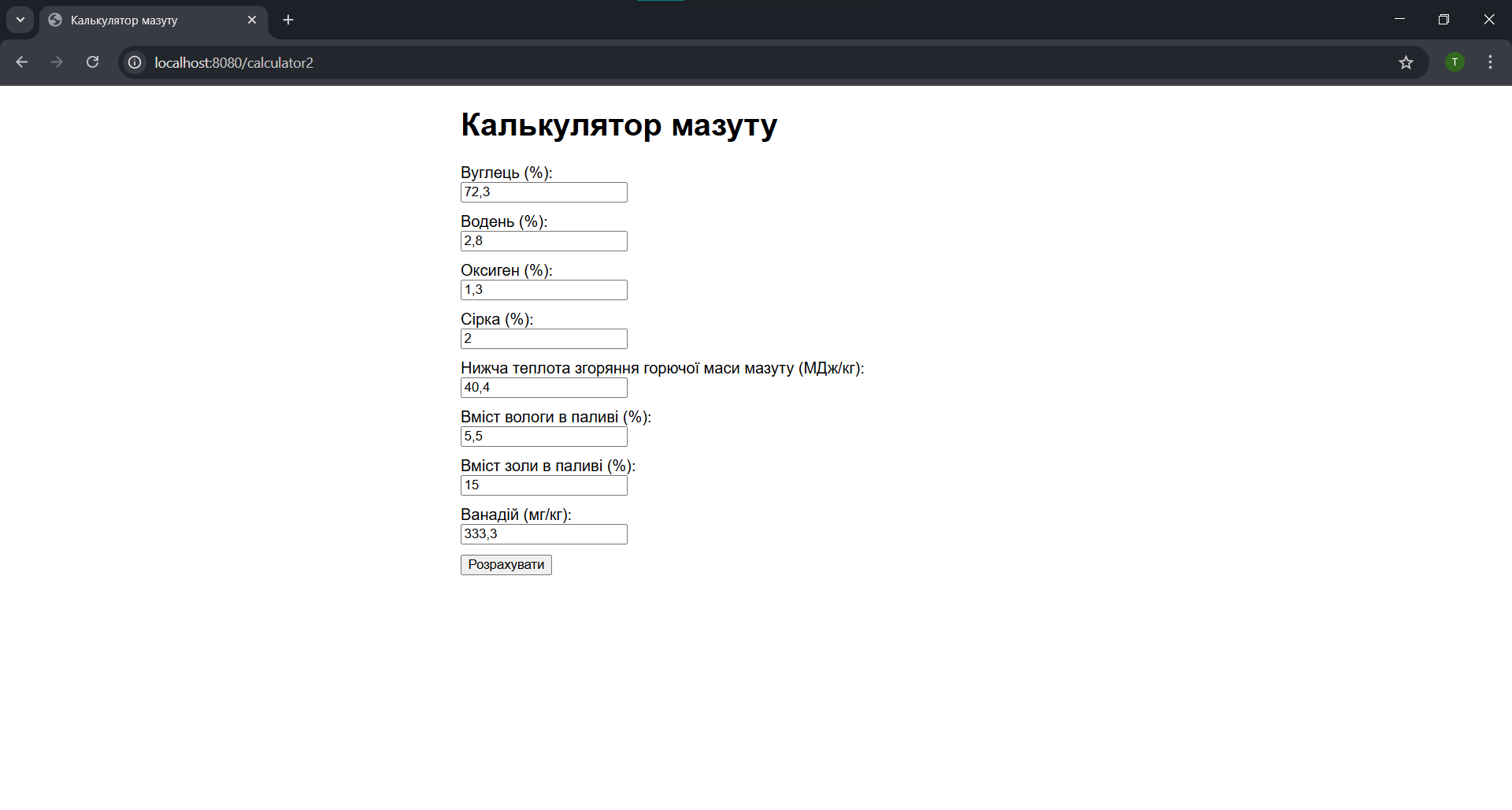
        data.Calculated = true

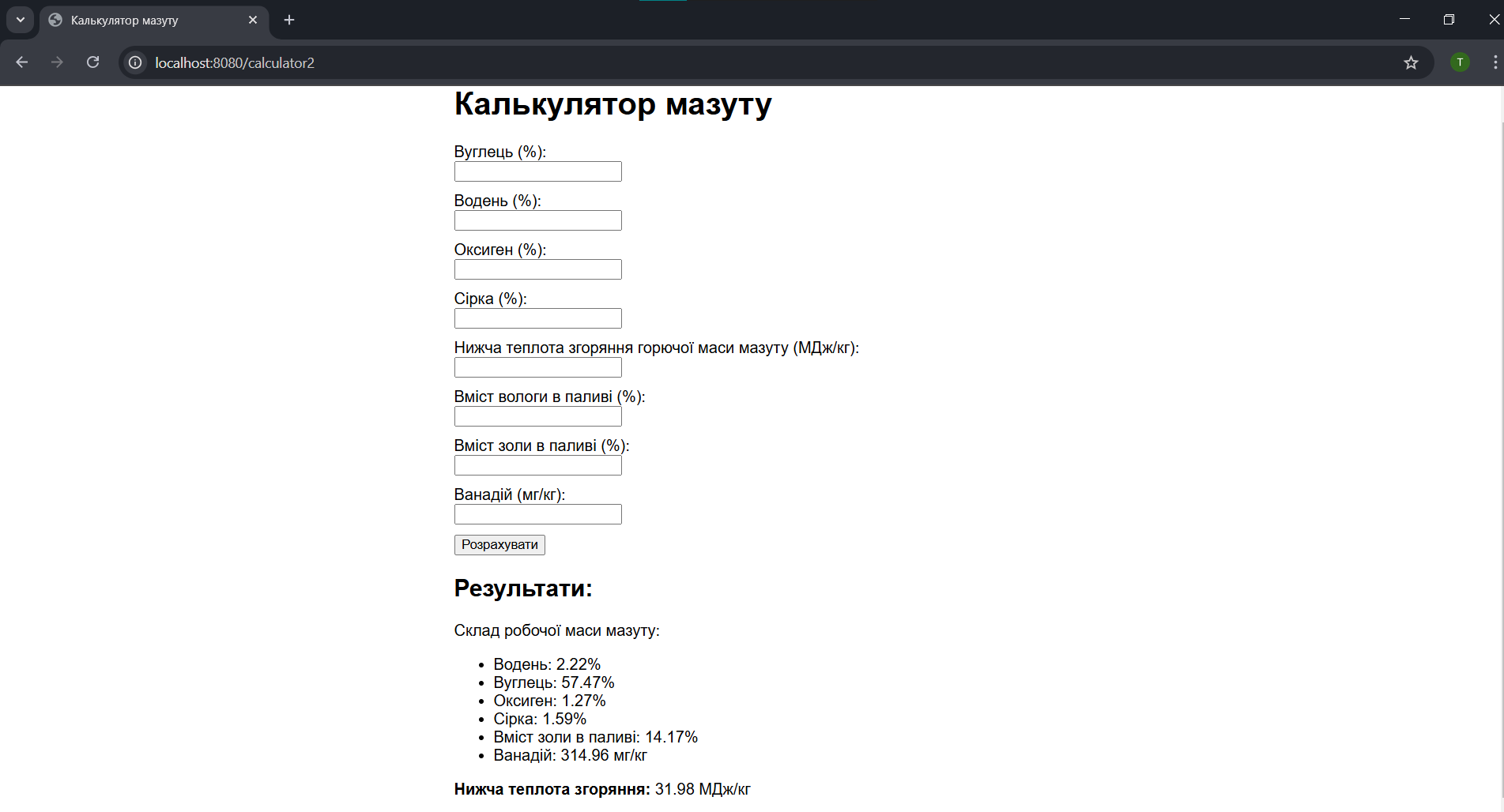
    }

    tmpl.Execute(w, data)

}

Сторінка, де вводитися, дані та виводитися фінальні результати наших розрахунків для другого калькулятора:





**Код програми для розрахунків:**

package calculators

import (

    // "fmt"

    "html/template"

    // "log"

    "net/http"

    "strconv"

)

// Структура для збереження даних форми та результатів

type Calculation struct {

    Carbon          float64

    Hydrogen        float64

    Oxygen          float64

    Sulfur          float64

    OilHeat         float64

    FuelMoisture    float64

    Ash             float64

    Vanadium        float64

    CarbonWM        float64

    HydrogenWM      float64

    OxygenWM        float64

    SulfurWM        float64

    AshWM           float64

    VanadiumWM      float64

    LowerHeatResult float64

    Calculated      bool

}

// func calculator2() {

//  http.HandleFunc("/", CalculatorHandler2)

//  http.ListenAndServe(":8080", nil)

// }

// Головна сторінка

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

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

    if err != nil {

        http.Error(w, "Error loading template", http.StatusInternalServerError)

        return

    }

    // Ініціалізація структури

    data := Calculation{}

    if r.Method == http.MethodPost {

        // Отримання даних з форми

        data.Carbon = parseInput(r.FormValue("carbon"))

        data.Hydrogen = parseInput(r.FormValue("hydrogen"))

        data.Oxygen = parseInput(r.FormValue("oxygen"))

        data.Sulfur = parseInput(r.FormValue("sulfur"))

        data.OilHeat = parseInput(r.FormValue("oilHeat"))

        data.FuelMoisture = parseInput(r.FormValue("fuelMoisture"))

        data.Ash = parseInput(r.FormValue("ash"))

        data.Vanadium = parseInput(r.FormValue("vanadium"))

        // Розрахунки

        factor1 := (100 - data.FuelMoisture - data.Ash) / 100

        factor2 := (100 - data.FuelMoisture/10 - data.Ash/10) / 100

        factor3 := (100 - data.FuelMoisture) / 100

        data.CarbonWM = formatValue(data.Carbon \* factor1)

        data.HydrogenWM = formatValue(data.Hydrogen \* factor1)

        data.OxygenWM = formatValue(data.Oxygen \* factor2)

        data.SulfurWM = formatValue(data.Sulfur \* factor1)

        data.AshWM = formatValue(data.Ash \* factor3)

        data.VanadiumWM = formatValue(data.Vanadium \* factor3)

        data.LowerHeatResult = formatValue(data.OilHeat\*factor1 - 0.025\*data.FuelMoisture)

        data.Calculated = true

    }

    tmpl.Execute(w, data)

}

// Функція для парсингу значень

func parseInput(value string) float64 {

    if val, err := strconv.ParseFloat(value, 64); err == nil {

        return val

    }

    return 0.0

}

// Функція для форматування значень

func formatValue(value float64) float64 {

    return float64(int(value\*100)) / 100

}

**Висновок:**

У результаті виконання практичної роботи №1 було реалізовано калькулятор, функція якого складається з розрахунку теплоти згоряння палива за допомогою мови програмування Go. Дані передавались з HTML-форми та результати виводились на тій же сторінці.