Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 4

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 № 4**

**Завдання:**

Створіть вебзастосунок для розрахунку струму трифазного КЗ, струму

однофазного КЗ, та перевірки на термічну та динамічну стійкість у складі:

1. Вибрати кабелі для живлення двотрансформаторної підстанції системи внутрішнього електропостачання підприємства напругою 10 кВ;

2. Визначити струми КЗ на шинах 10 кВ ГПП;

3. Визначити струми КЗ для підстанції Хмельницьких північних електричних мереж (ХПнЕМ), яка може мати три режими: нормальний режим; мінімальний режим; аварійний режим (див. Приклад 7.4.).

**Хід виконання:**

Основні формули для роботи калькулятора були задані у практичному матеріалі

**Код програми:**

package main

import (

    "fmt"

    "log"

    "math"

    "net/http"

    "strconv"

    "text/template"

)

type DataCable struct {

    BronC      string

    AabC       string

    Calculated bool

}

type DataKZ1 struct {

    Kz1        string

    Calculated bool

}

type DataKZ2 struct {

    I3Normal  string

    I3Min     string

    I2Normal  string

    I2Min     string

    DI3Normal string

    DI3Min    string

    DI2Normal string

    DI2Min    string

    Calculated bool

}

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

    tmpl := template.Must(template.ParseFiles("templates/calculators/calculator1.html"))

    if r.Method == http.MethodPost {

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

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

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

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

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

        Im := Sm / (2 \* math.Sqrt(3.0) \* U)

        bron := Im / JEk

        aab := I \* math.Sqrt(Time) / 92.0

        data := DataCable{

            BronC:      fmt.Sprintf("%.2f", bron),

            AabC:       fmt.Sprintf("%.2f", aab),

            Calculated: true,

        }

        tmpl.Execute(w, data)

    } else {

        tmpl.Execute(w, nil)

    }

}

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

    tmpl := template.Must(template.ParseFiles("templates/calculators/calculator2.html"))

    if r.Method == http.MethodPost {

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

        Uc := 10.5

        Kz1 := Uc / (math.Sqrt(3.0)\*(Uc\*Uc/Kzu) + ((Uc / 100) \* (Uc \* Uc / 6.3)))

        data := DataKZ1{

            Kz1:        fmt.Sprintf("%.2f", Kz1),

            Calculated: true,

        }

        tmpl.Execute(w, data)

    } else {

        tmpl.Execute(w, nil)

    }

}

func CalculatorHandler3(w http.ResponseWriter, r \*http.Request) {

    tmpl := template.Must(template.ParseFiles("templates/calculators/calculator3.html"))

    if r.Method == http.MethodPost {

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

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

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

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

        U := 115.0

        Ub := 11.0

        sqrt3 := math.Sqrt(3.0)

        multiplier := math.Pow(10, 3)

        Xt := (11.1 \* math.Pow(U, 2)) / (100 \* 6.3)

        Z := math.Sqrt(math.Pow(Rh, 2) + math.Pow(Xh+Xt, 2))

        ZMin := math.Sqrt(math.Pow(Rm, 2) + math.Pow(Xm+Xt, 2))

        I3Normal := (U \* multiplier) / (sqrt3 \* Z)

        I3Min := (U \* multiplier) / (sqrt3 \* ZMin)

        I2Normal := I3Normal \* (sqrt3 / 2)

        I2Min := I3Min \* (sqrt3 / 2)

        k := math.Pow(Ub, 2) / math.Pow(U, 2)

        ZTrue := math.Sqrt(math.Pow(Rh\*k, 2) + math.Pow((Xh+Xt)\*k, 2))

        ZMinTrue := math.Sqrt(math.Pow(Rm\*k, 2) + math.Pow((Xm+Xt)\*k, 2))

        DI3Normal := (Ub \* multiplier) / (sqrt3 \* ZTrue)

        DI3Min := (Ub \* multiplier) / (sqrt3 \* ZMinTrue)

        DI2Normal := DI3Normal \* (sqrt3 / 2)

        DI2Min := DI3Min \* (sqrt3 / 2)

        data := DataKZ2{

            I3Normal:   fmt.Sprintf("%.2f", I3Normal),

            I3Min:      fmt.Sprintf("%.2f", I3Min),

            I2Normal:   fmt.Sprintf("%.2f", I2Normal),

            I2Min:      fmt.Sprintf("%.2f", I2Min),

            DI3Normal:  fmt.Sprintf("%.2f", DI3Normal),

            DI3Min:     fmt.Sprintf("%.2f", DI3Min),

            DI2Normal:  fmt.Sprintf("%.2f", DI2Normal),

            DI2Min:     fmt.Sprintf("%.2f", DI2Min),

            Calculated: true,

        }

        tmpl.Execute(w, data)

    } else {

        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, "Internal Server Error", http.StatusInternalServerError)

        return

    }

    tmpl.Execute(w, nil)

}

func main() {

    http.HandleFunc("/", homePage)

    http.HandleFunc("/calculator1", CalculatorHandler1)

    http.HandleFunc("/calculator2", CalculatorHandler2)

    http.HandleFunc("/calculator3", CalculatorHandler3)

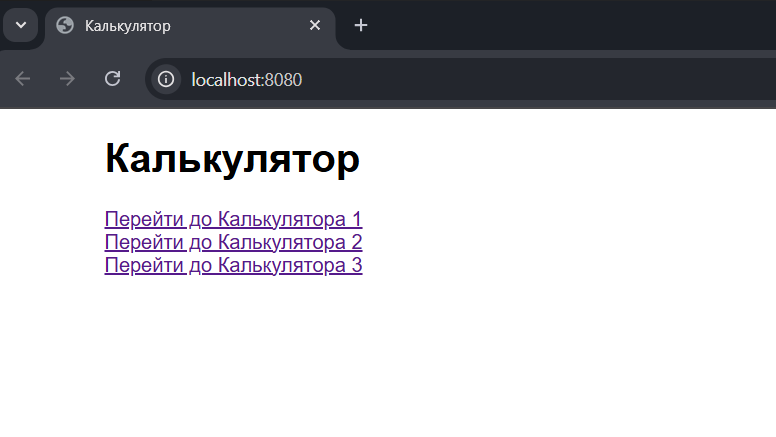
    port := ":8080"

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

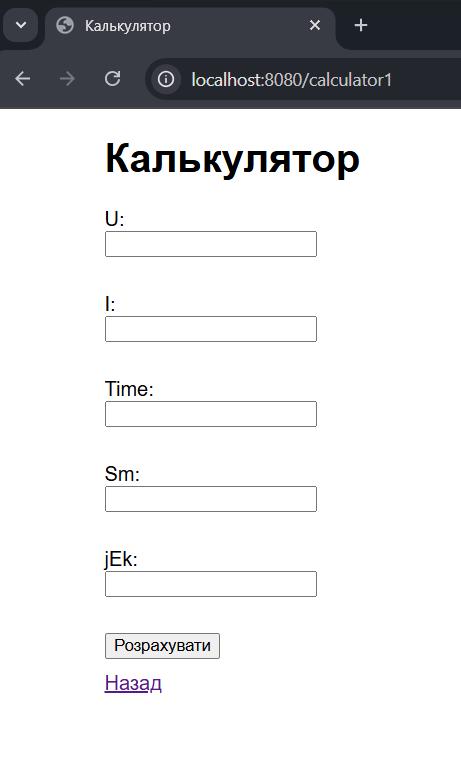
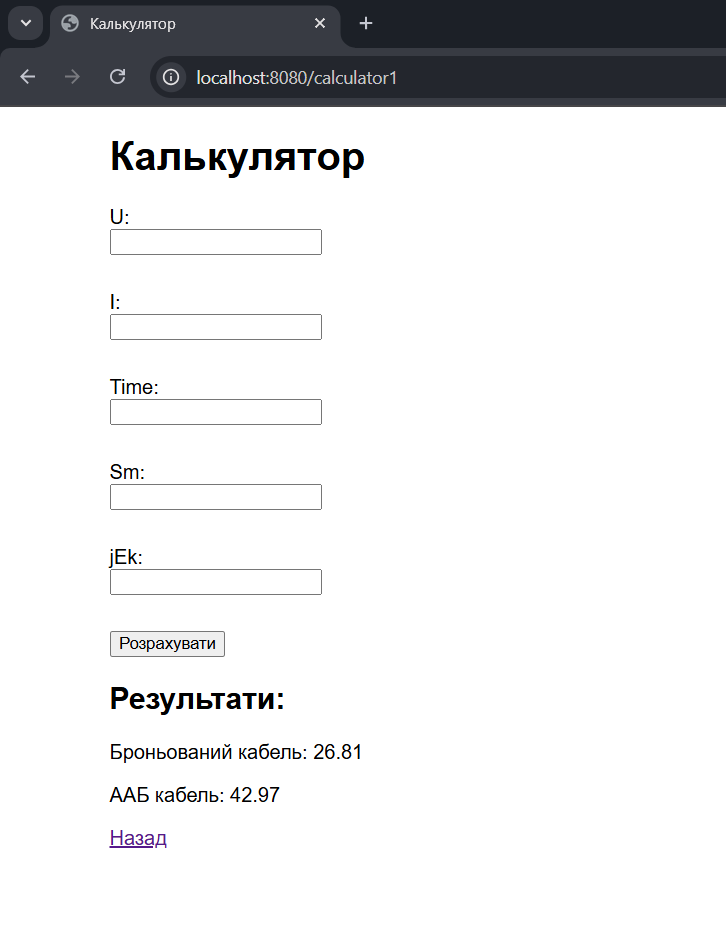
    log.Fatal(http.ListenAndServe(port, nil))

}

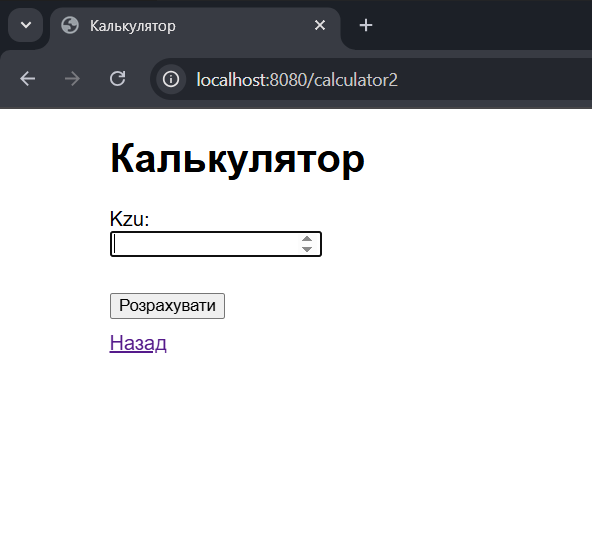
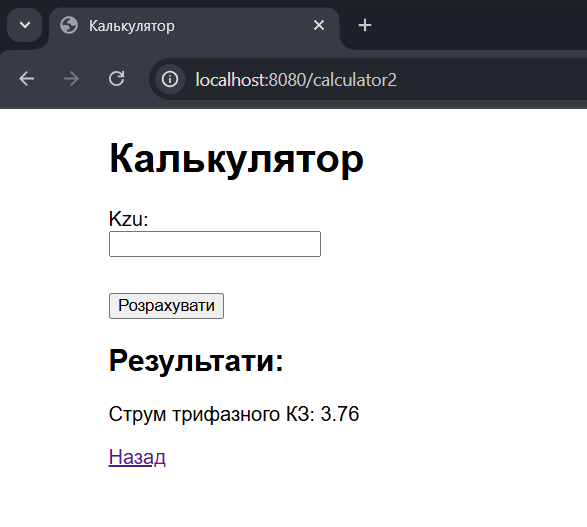
Головна сторінка



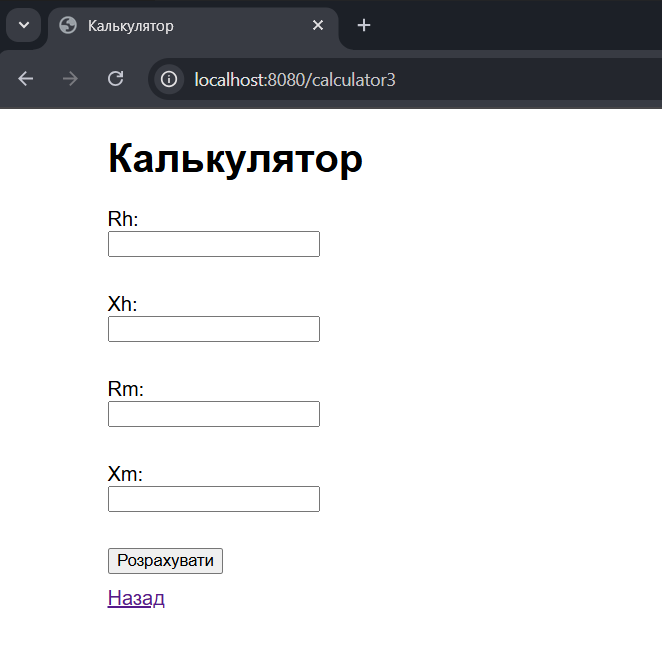
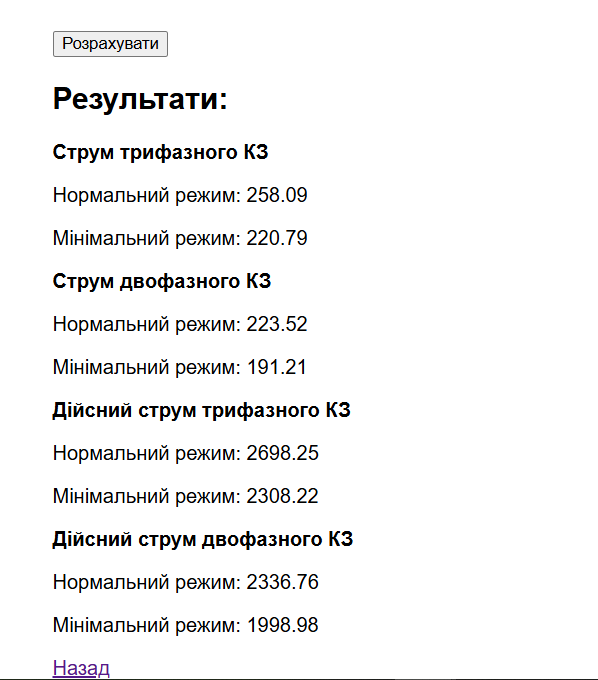
Калькулятор для вибору кабелів живлення двотрансформаторної підстанції системи внутрішнього електропостачання підприємства напругою 10 кВ

Калькулятор для визначення струми КЗ на шинах 10 кВ ГПП

Калькулятор для визначення струми КЗ для підстанції Хмельницьких північних електричних мереж (ХПнЕМ), яка може мати три режими: нормальний режим; мінімальний режим; аварійний режим

**Висновок**

В результаті виконання лабораторної роботи №4 було створено калькулятори мовою програмування Go для розрахунку струму трифазного КЗ, струмуоднофазного КЗ, та перевірки на термічну та динамічну стійкість у складі.