Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут  ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Практична робота № 5

з курсу: «Програмування веб застосунків»

Виконала:  
студентка 4-го курсу,  
групи ТВ-13  
Рябець Катерина Олександрівна

 Посилання на GitHub репозиторій: <https://github.com/KateRiabets/Go>

**Перевірив:**

Недашківський О.Л.

Київ 2025

1. **Теоретичний матеріла**

**Надійність** – це здатність ЕПС виконувати задані функції з дотриманням нормативних експлуатаційних показників.

**Відмова** – це втрата працездатності окремих елементів, частини чи всієї системи.

Відмови бувають :

* Повна або часткова
* Раптова або поступова.
* Стійка або нестійка.
* Явна або прихована.
* Конструктивна, технологічна чи експлуатаційна.

**Відновлення – це повернення працездатності шляхом ремонту чи заміни обладнання.**

Теоретичний показник – **ймовірність безвідмовної роботи за час t (р(t))**.

До практичних показників відносять:

* Частота відмов (ω).
* Частота ремонтів (µ).
* Тривалість відновлення (tв).
* Тривалість поточного ремонту (tр).

ЛЕП – найуразливіший елемент ЕПС. На лінії електропередачі (ЛЕП) припадає 85–95% усіх вимкнень. Основні причини пошкоджень ЛЕП: грозові перекриття ізоляції, ожеледь, вітер, вібрація проводів, падіння дерев, контакт повітряних ліній із будівельними машинами.

До основних причин пошкоджень кабельних ліній(КЛ) відносять: будівельні роботи,(пошкодження технікою), старіння ізоляції. попадання вологи або механічні пошкодження ізоляції (гризуни, пробої).

Таблиця 1.1 – Показники надійності ліній електропередачі

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Елемент ЕПС | ω, рік-1 на 1 км | tв, год | µ, рік-1  на 1 коло | tп, год. |
| ПЛ-110 кВ | 0,007 | 10 | 0,167 | 35 |
| ПЛ-35 кВ | 0,02 | 8 | 0,167 | 35 |
| ПЛ-10 кВ | 0,02 | 10 | 0,167 | 35 |
| КЛ-10 кв (траншея) | 0,03 | 44 | 1 | 9 |
| КЛ-10 кВ (кабельний канал) | 0,005 | 17,5 | 1 | 9 |

Основне про причини відмов силових трансформаторів

* Порушення ізоляції обмоток (зовнішні/внутрішні перенапруги, наскрізні струми КЗ).
* Перевантаження та старіння.
* Дефекти виготовлення.
* Пошкодження регулювальних пристроїв і контактних з'єднань.
* Зниження рівня трансформаторної оливи.
* Пошкодження вводів через перекриття ізоляції.

Основне про причини відмов вимикачів

* Несправність приводів.
* Обгорання контактів.
* Зношення дугогасильних камер.
* Заводські дефекти.
* Помилки персоналу при перемиканнях.

Основне про причини однофазних замикань на землю

* Горіння заземлювальних дуг через недостатню компенсацію ємнісних струмів.
* Наслідки: перенапруги, пробої ізоляції, руйнування ізоляторів.

Таблиця 1.2 – Показники надійності електрообладнання підстанцій

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Елемент ЕПС | ω, рік-1 на 1 км | tв, год | µ, рік-1  на 1 коло | tп, год. |
| Т-110 кВ | 0,015 | 100 | 1 | 43 |
| Т-35 кВ | 0,02 | 80 | 1 | 28 |
| Т-10 кВ (кабельна мережа 10 кВ) | 0,005 | 60 | 0,5 | 10 |
| Т-10 кВ (повітряна мережа 10 кВ) | 0,05 | 60 | 0,5 | 10 |
| В-110 кВ (елегазовий) | 0,01 | 30 | 0,1 | 30 |
| В-10 кВ (малооливний) | 0,02 | 15 | 0,33 | 15 |
| В-10 кВ (вакуумний) | 0,01 | 15 | 0,33 | 15 |
| Збірні шини 10 кВ на 1 приєднання | 0,03 | 2 | 0,167 | 5 |
| АВ-0,38 кВ | 0,05 | 4 | 0,33 | 10 |
| ЕД 6, 10 кВ | 0,1 | 160 | 0,5 | - |
| ЕД 0,38 кВ | 0,1 | 50 | 0,5 | - |

Частота відмов одноколової системи (одного кола системи) розраховується за формулою:

, (1.1)

а середня тривалість відновлення –

, (1.2)

де

ωі – частота відмов окремих елементів;

tві – тривалість відновлення окремих елементів.

Для одноколової системи (без резервування) коефіцієнт аварійного простою (неготовності) розраховується за формулою:

, (1.3)

де

ωос – частота відмов одного кола системи , рік –1 ;

tв.ос – середня тривалість її відновлення в роках (якщо тривалість відновлення задана в годинах, то її необхідно поділити на 8760 год.)

Коефіцієнт планового простою одноколової системи розраховується за формулою:

, (1.4)

де

kп.і = µіtпі ,– значення коефіцієнта планового простою і-го елемента;

µі – частота планових ремонтів елементів;

tпі – тривалість планового ремонту елементів в роках (якщо тривалість планового ремонту задана в годинах, то її необхідно поділити на 8760 год.).

У разі паралельного з’єднання двох кіл частоту одночасної їх відмови знаходять за виразом

, (1.5)

а середню тривалість відновлення –

, (1.6)

Частота відмови двоколової системи з урахуванням секційного вимикача

, (1.7)

де ωcв – частота відмов секційного вимикача

Коефіцієнт аварійного простою (неготовності) двоколової системи

, (1.8)

Математичне сподівання збитків від переривання електропостачання може бути визначене за виразом

, (1.9)

де

Зпер(0) – прямі збитки від самого факту переривання електропостачання;

З(t) – прямі і додаткові збитки, які залежать від тривалості перерви електропостачання;

ω – частота відмов;

tВ – тривалість відновлення після відмови.

Збитки від переривання електропостачання можна оцінити за спрощеною формулою

, (1.10)

де Зпер.а, Зпер.п − питомі складові збитків від, відповідно, аварійного і планового недовідпущень електроенергії; М(Wнед.а), М(Wнед.п) − математичні сподівання середньорічних аварійного і планового недовідпущень електроенергії

Математичне сподівання аварійного і планового недовідпущень електроенергії знаходять з формул:

, (1.11)

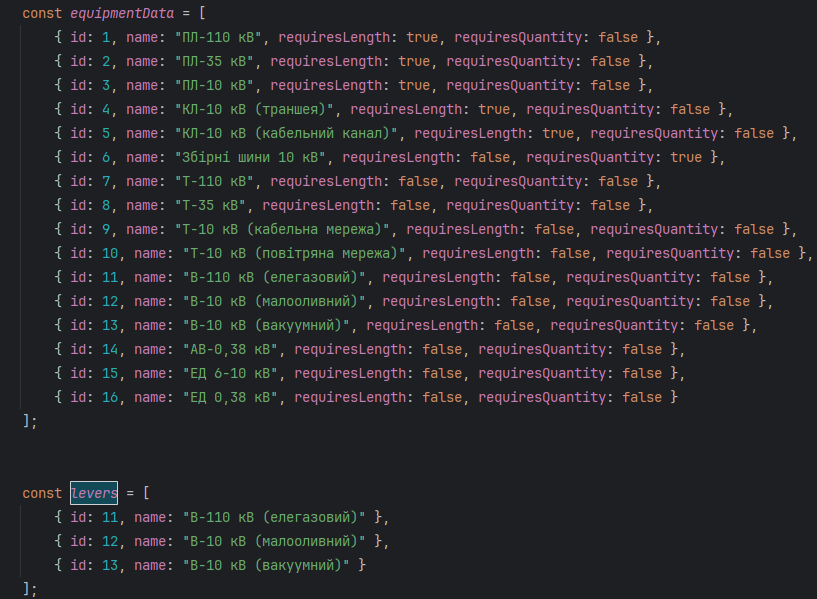
, (1.12)

де kа, kп − коефіцієнти аварійного і планового простоїв.

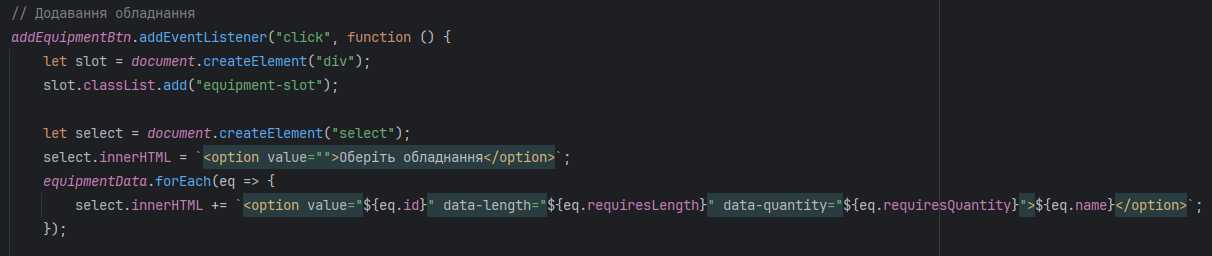
1. **Опис програмної реалізації**

**Завдання 1**

При виконанні цього завдання було вирішено реалізувати спілкування з клієнтом через json. Таким чином на клієнті було створено перелік всього обладнання та перелік вимикачів (для двоколової системи доступний вибір тільки вимикача).



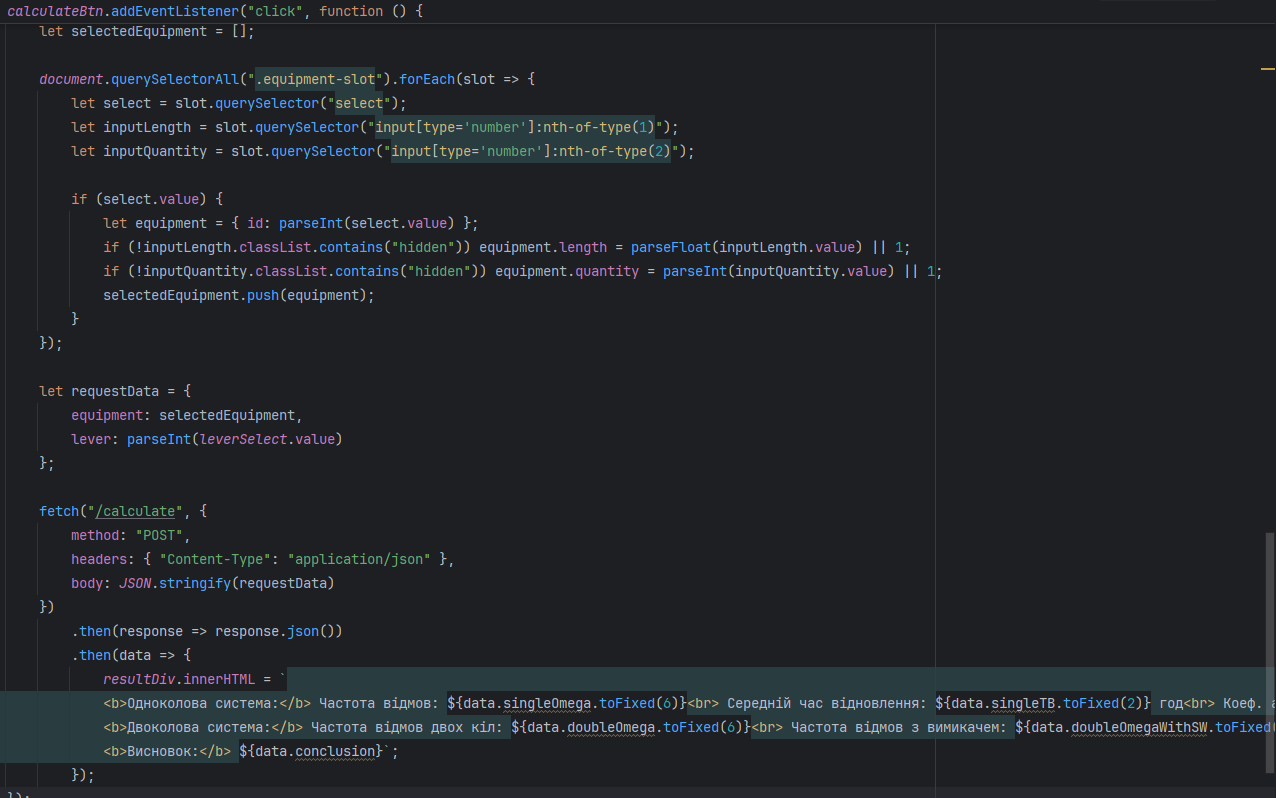
Коли користувач натискає кнопку "Додати обладнання", створюється новий блок div. Створюється список, де можна вибрати обладнання, в який додаються опції з equipmentData. Для одноколової системи можна обирати будь яке обладнання з таблиці і в будь якій кількості. Для двоколової системи вважаємо, що вона складається з 2 ідентичних одноколових та вимикача, тому можливий вибір лише його. Також додавання деякого обладнання потребує введення кількості (шини) або довжини (лінії передач).



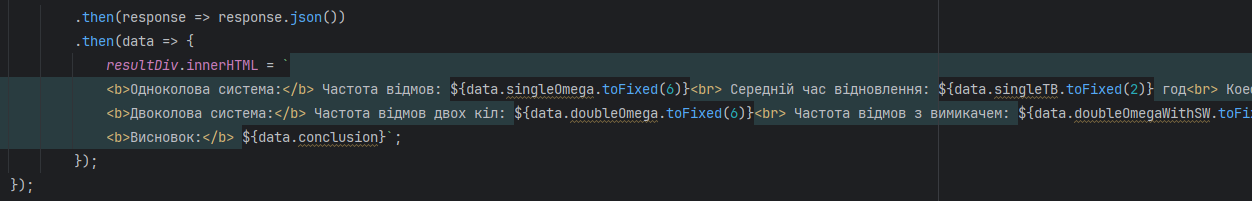
В залежності від параметрів requiresLength та requiresQuantity показуються поля для введення довжини та кількості (за замовчуванням вони приховані).



При натисканні ні кнопку «Розрахувати» починається збір даних. Шукаються всі .equipment-slot (вибране обладнання),зчитуються ID, довжина та кількість (якщо вони потрібні) та зберігаються у selectedEquipment. Потім створюється JSON-об'єкт для відправки на сервер. JSON вілправляється на сервер (/calculate) POST запитом.



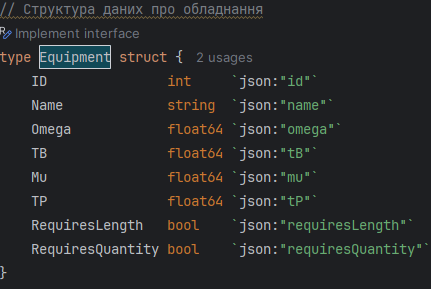
Потім отримується відповідь від сервера та показується результат.



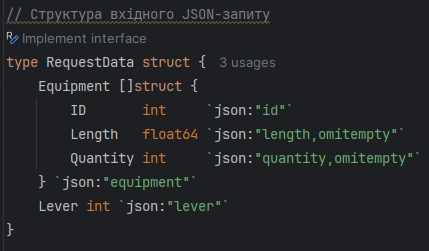
Тепер щодо серверу.

У програмі використовуються кілька структур.

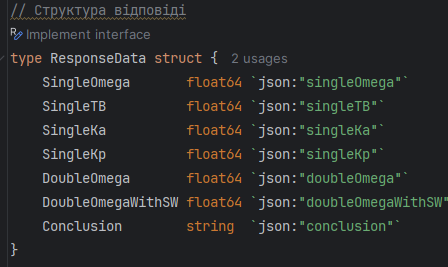
Equipment – описує всі параметри обладнання.



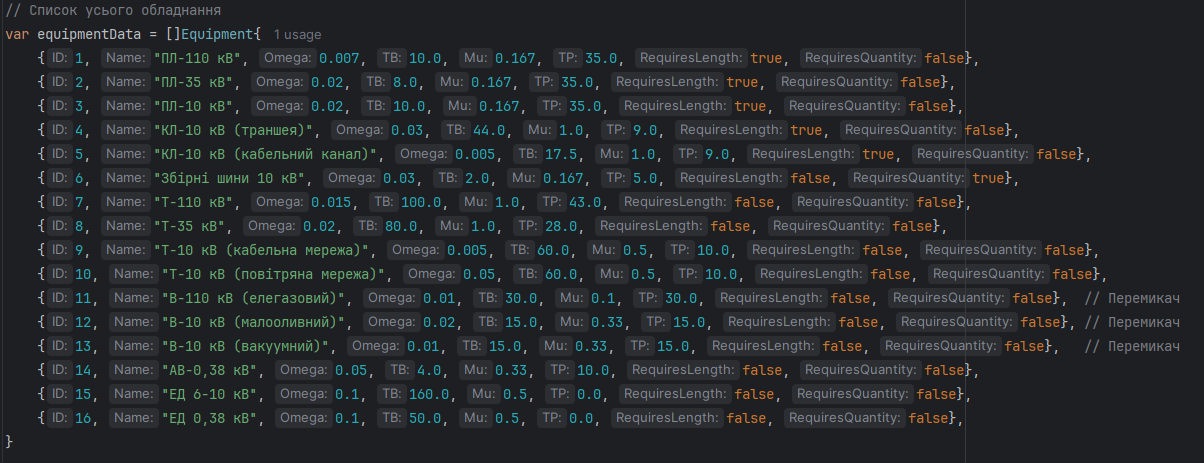
RequestData – містить список вибраного обладнання та перемикач для двоколової системи.



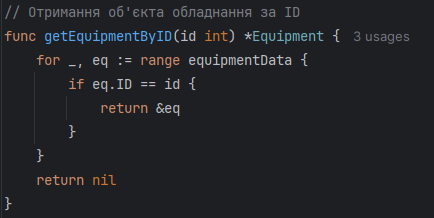
ResponseData – використовується для повернення результатів у JSON-відповіді.



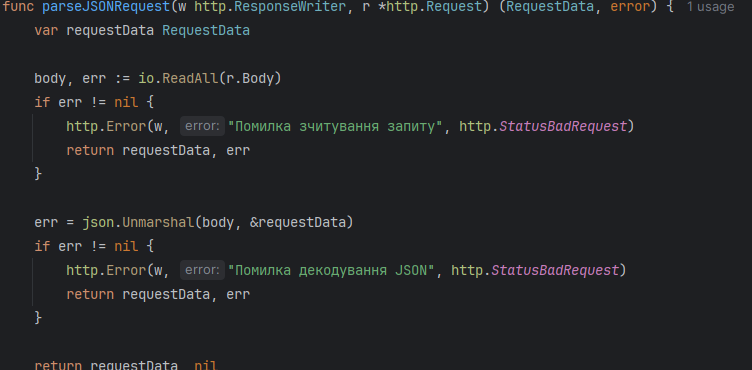
equipmentData – це глобальна змінна, яка містить усі доступні види обладнання і їхні характеристики.



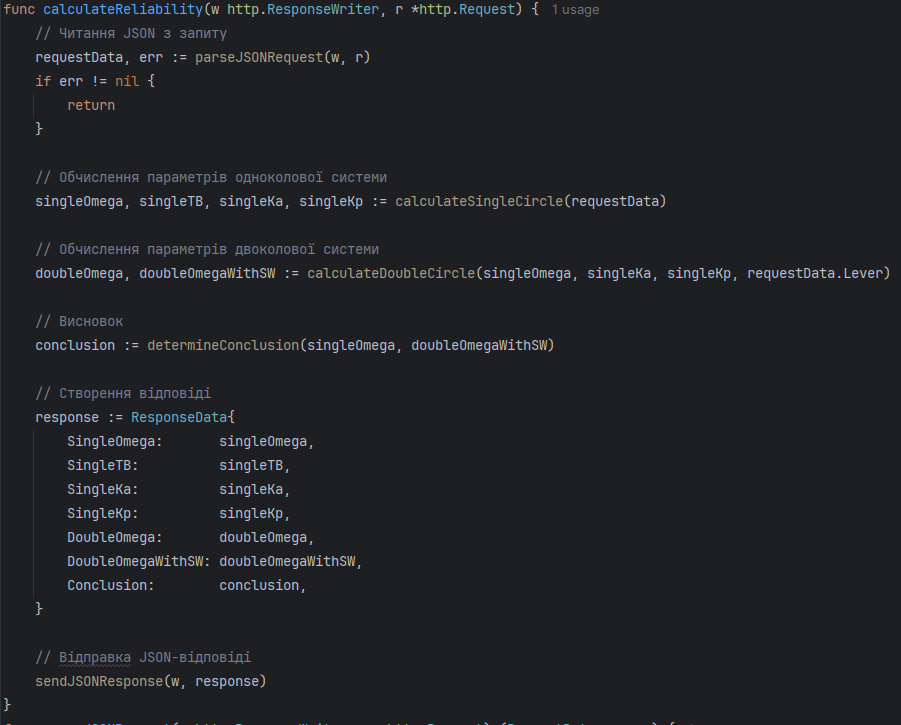
getEquipmentByID функція шукає об'єкт обладнання за його ID, яке було передано через JSON в equipmentData і повертає його. Якщо нічого не знайдено, повертається nil.



Обробка JSON запиту від клієнта відбувається в функції parseJSONRequest. Ця функція зчитує тіло HTTP-запиту (io.ReadAll). Декодує JSON у RequestData за допомогою json.Unmarshal. Якщо сталася помилка – відправляє http.Error.



Функція calculateReliability викликає parseJSONRequest, щоб отримати дані від клієнта. Виконує обчислення для одноколової системи (функція calculateSingleCircle). Обчислює параметри для двоколової системи (функція calculateDoubleCircle). Формує висновок (функція determineConclusion). Відправляє JSON-відповідь клієнту через sendJSONResponse.



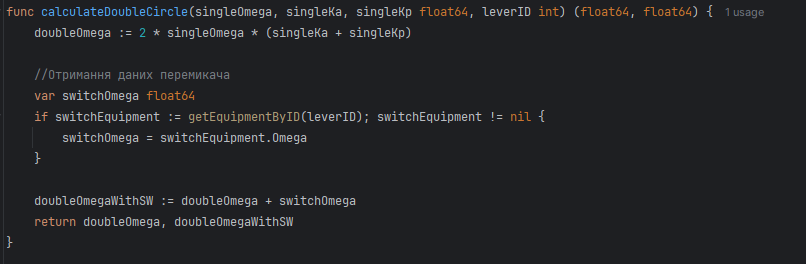
Функція calculateSingleCircle відповідає за розрахунки для одноколовох системи.



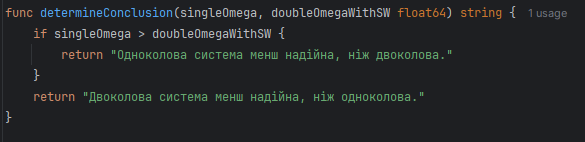
Розрахунки для двоколової системи виконує функція calculateDoubleCircle.

Розрахунки для двоколової системи виконуємо як

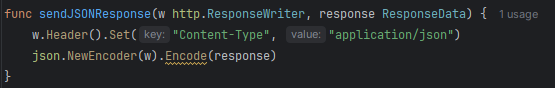
Оскільки маємо справу по суті з 2 ідентичними одноколовими та вимикачем.



Висновок робиться на основі порівняння та

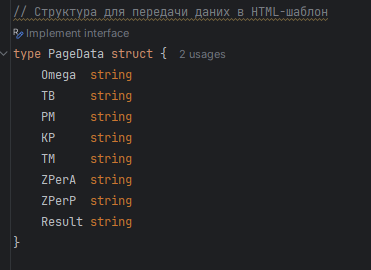


Функція sendJSONResponse перетворює ResponseData у JSON та відправляє його клієнту.



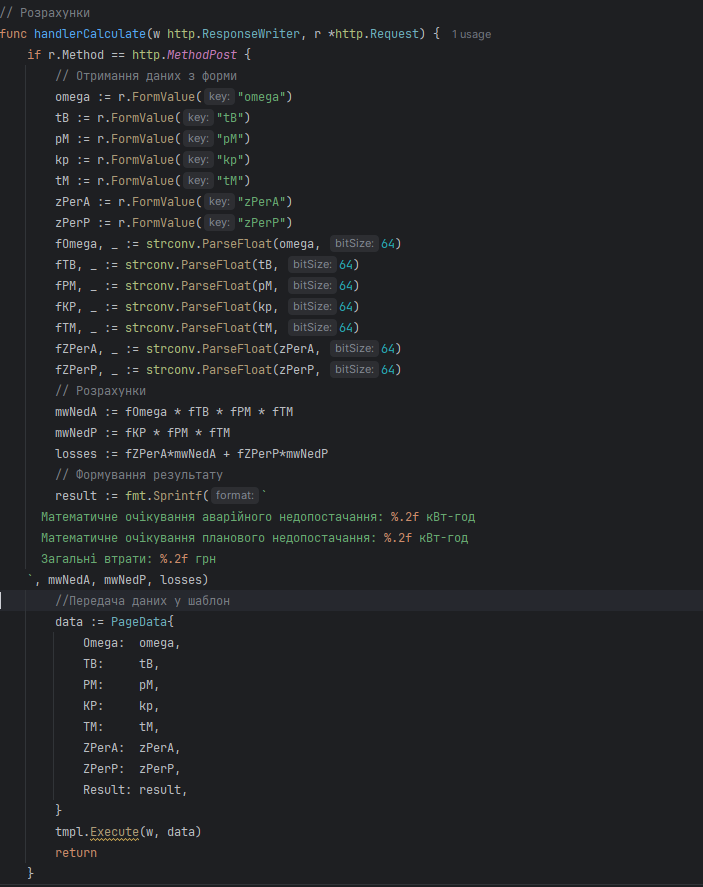
**Завдання 2**

PageData – це структура, яка містить змінні для передачі даних у HTML-шаблон.

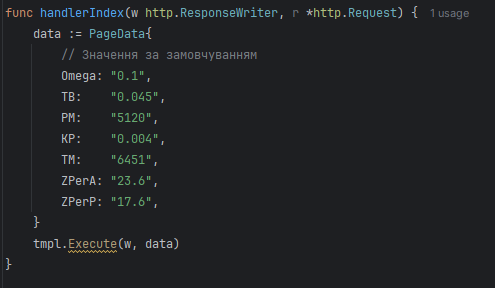


При натисканні на кнопку «Розрахувати» відбуваються обчислення та формування результату. Розрахунки математичне сподівання аварійного і планового недопостачання електроенергії відбуваються за формулами 1.11 та 1.12 відповідно. Збитки розраховуємо за формулою 1.10.

Функція r.FormValue("omega") отримує значення з HTML-форми за ім'ям поля. strconv.ParseFloat() використовується, щоб перетворити текстові дані у float64.

****

Функція handlerIndex(w http.ResponseWriter, r \*http.Request). Ця функція обробляє запит на головну сторінку (/). Змінна data, яка містить значення за замовчуванням. tmpl.Execute(w, data) виконує рендеринг HTML-шаблону template.html, підставляючи дані.



1. **Результати перевірки на контрольному прикладі**

**Завдання 1**

|  |  |
| --- | --- |
|  |  |

**Завдання 2**

|  |  |
| --- | --- |
|  |  |

У розрахунках з контрольного прикладу було помічено неточності. Додаю перевірку зазначених розрахунків на калькуляторі.





**Висновок:**

У результаті виконання практичної роботи було створено калькулятор для порівняння надійності одноколової та двоколової систем електропередачі та розрахунку збитків від перерв електропостачання у разі застосування однотрансформаторної ГТП. Було реалізовано користувацький інтерфейс та, власне, розрахунки. Правильність роботи калькулятора було протестовано на контрольних прикладах. Також в ході виконання було помічено деякі неточності в розрахунках у наданому матеріалі (контрольних прикладах). Додатково розрахунки було перевірено за допомогою калькулятора Photomath. З урахуванням цього, можна стверджувати, що створений калькулятор працює коректно.

Робота з JSON у Go потребує ручного парсингу (json.Unmarshal), тоді як у Kotlin досить бібліотек типу Gson чи Kotlinx.serialization, і все автоматично. У Go (Завдання 1) було реалізовано окрему функціюparseJSONRequest(w, r), що читає io.ReadAll(r.Body) і парсить json.Unmarshal(...). У Kotlin такого немає, бо data class Equipment(...) можна напряму десеріалізувати через Gson або Kotlinx.serialization. У Go весь UI — це окремий HTML-файл. У Kotlin використовується Jetpack Compose, де UI будується динамічно через Composable-функції.

У Go (Завдання 2) отримання даних із форми відбувається через r.FormValue("omega"), а потім значення парсяться у float64 через strconv.ParseFloat(). У Kotlin всі значення зберігаються у змінних remember { mutableStateOf("") }, а їх конвертація у Double відбувається безпосередньо перед розрахунком. У Go результат формується через fmt.Sprintf() і передається у PageData, який рендериться у HTML-шаблоні. Тобто кожен новий результат — це повне перезавантаження сторінки. У Kotlin результат просто змінюється у remember { mutableStateOf("") }, і UI миттєво оновлюється без перезавантаження.