

ДАТА ЗВІТУ: 2019-12-10 12:02:00

ЗАГОЛОВОК:

Зітинюк_Ю_МКР_ГДІГ_21.docx

АВТОР:

Зітинюк Ю.

ПІДРОЗДІЛ:

NULP2

ПРОПУЩЕНІ ВЕБ-СТОРИНКИ:



НАУКОВИЙ КЕРІВНИК:

ІГД Кафедра

ДАТА ЗАВАНТАЖЕННЯ ФАЙЛУ:

2019-12-10 12:00:00

Рівень запозичень

Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

0,00%

КП 1



% Слів знайдених поза Бази юридичних актів методом
перевірки кожні 5 слів

25
Довжина фрази для коефіцієнта
подібності

0,00%

КП 2



% Слів знайдених поза Бази юридичних актів методом
перевірки кожні 25 слів

3871
Кількість слів

1,29%

КЦ



% цитат

27747
Кількість символів

Тривога

У цьому розділі Ви можете знайти інформацію щодо модифікації тексту, яка може бути спрямована на зміну результатів аналізу. Невидимі для особи, яка оцінює вміст документа у роздруковці чи файлі, вони впливають на фрази, порівнювані під час аналізу тексту (викликаючи передбачувані помилки), щоб приховати запозичення, а також підробити значення у звіті про подібність. Слід оцінити, чи є модифікації навмисними чи ні.

Тривога	2
кількість символів з інших алфавітів може означати спробу обману, уважно перевірте!	
Інтервали	0
кількість збільшених відстаней між літерами - будь ласка, перевірте, чи вони імітують пробіли, викликаючи приєднання слів до звіту	
Мікропробіли	0
кількість пробілів із нульовою довжиною - будь ласка, перевірте, чи вони розміщені всередині слів та спричинили поділ слів у тексті	
Білий знаки	0
кількість символів з білим кольором шрифту - будь ласка, перевірте, чи використовуються вони замість пробілів, викликаючи приєднання слова (у звіті колір літер змінено на чорний, щоб показати їх)	

Активні списки запозичень

Прокручайте список та аналізуйте, особливо, фрагменти, які перевищують КП 2 (позначено жирним шрифтом). Скористайтесь посиланням "Позначити фрагмент" та перегляньте, чи є вони короткими фразами, розкиданими в документі (випадкові схожості), численними короткими фразами поруч з іншими (мозаїчний плагіат) або великими фрагментами без зазначення джерела (прямий плагіат).

10 найдовших фраз (0,00 %)

Десять найдовших фрагментів знайдений у всіх доступних ресурсах.

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	АВТОР	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ
НЕМАЄ ПОДІБНОСТЕЙ			

з бази даних RefBooks (0,00 %)

Всі фрагменти знаходяться в базі даних RefBooks, яка містить більше 3 мільйонів текстів від редакторів і авторів.

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	АВТОР	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (КІЛЬКІСТЬ ФРАГМЕНТІВ)
НЕМАЄ ПОДІБНОСТЕЙ			

з домашньої бази даних (0,00 %)

Всі фрагменти знайдені у внутрішній базі даних вашої установи.

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	АВТОР	ДАТА ІНДЕКСАЦІЇ	ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
НЕМАЄ ПОДІБНОСТЕЙ				

з програми обміну базами даних (0,00 %)

Всі фрагменти знайдені в базі даних інших установ.

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК НАЗВА БАЗИ ДАНИХ	АВТОР	ДАТА ІНДЕКСАЦІЇ	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (КІЛЬКІСТЬ ФРАГМЕНТІВ)
---------------------	-------------------------------	-------	--------------------	--

НЕМАЄ ПОДІБНОСТЕЙ

з Інтернету (0,00 %)

Всі фрагменти, які були знайдені у відкритому доступі глобальних інтернет-ресурсів.

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
---------------------	-------------	---------------------------------

НЕМАЄ ПОДІБНОСТЕЙ

Вміст пошуку - позначення запозичень:

Будь ласка, зверніть увагу на те, що система не дає остаточної оцінки. Якщо виникають підозри, Звіт Подібності повинен бути переданий на ретельний аналіз.

ЗМІСТ

- - База даних університету
- - Джерело Інтернет
- - База даних RefBooks
- - Цитати

РОЗДІЛ 2. ДОСЛІДЖЕННЯ СТРУКТУРИ ФОРМАТНИХ ФАЙЛІВ ТА ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ІНТЕРАКТИВНОГО КОНВЕРТЕРА

2.1. Структура форматного файлу GSI

Формат GSI - один з перших форматів Leica, тому зараз його підтримують багато цифрових пристроїв Leica Geosystems: цифрові нівеліри, електронні тахеометри, супутникові приймачі. GSI - скорочення від Geo Serial Interface. Формат надає двосторонній інтерфейс між приладом та комп'ютером користувача. Розрізняють два його типи GSI-8 та GSI-16. Цифри 8 та 16 означають кількість цифр, доступних для запису величин.

Рис. 2.1. Зразок файлу GSI-8

Рис. 2.2. Зразок файлу GSI-16

Файл складається із стрічок довільної довжини. Кожна стрічка містить в собі інформацію про єдину точку. В свою чергу стрічка поділена на поля, розділені відступом. Кожне поле умовно поділяється на дві частини: метадані, значення.

Слід зазначити, що характер інформації, що знаходиться в полі залежить від її позиції в цьому полі. Для прикладу розглянемо структуру поля GSI-8:

Рис. 2.3. Запис точки в GSI-8

- Позиція 1-2: WI (Word Index) - індекс слова або заголовок поля. Цей індекс однозначно співвідноситься з таблицею індексів, поданих нижче (Таблиця 2.2). Значення індексу вказує на тип інформації, зашифрованої в полі (кут, віддаль, висота, координата, номер точки) На зразку це відповідно «11», «81», «82».
- Позиція 3-6: Інформація про формат даних в полі (таблиця 2.1). В першому полі це завжди номер стрічки в файлі. На зразку це відповідно «0002», «..00», «..00».
- Позиція 4: Знак «+» чи «-».
- Позиція 8-15: Значення. Якщо поле містить декілька значень, то вони розділені знаком. На зразку це відповідно «0000A113», «00005387», «00000992».
- Позиція 16: Відступ - сепаратор.

Інформація про формат даних в полі

Таблиця 2.1.

Позиція Пояснення Можливі значення Опис Доступність

3 Зазвичай, це сепаратор між індексом поля та інформацією про дані. У випадку, коли інформація знаходиться в першому полі вона набуває числового значення. "." сепаратор між індексом поля та інформацією про дані Для всіх полів

0-9 цифра в номері поточного рядка

4 Автоіндекс 0 Вимкнений Для кутів

1 Увімкнений

3 Увімкнений

5 Режим введення даних 0 Дані з інструмента Для вимірів

1 Вручну введені дані з клавіш

2 Обчислене значення з корекцією горизонтальних кутів

3 Обчислене значення без корекції горизонтальних кутів

4 Результат обчислений з функції

6 Одиниці вимірювання 0 Метри, остання цифра - 1мм Для вимірів

1 Фути, остання цифра - 1/1000фт

2 400 гон
3 360 десяткових градусів
4 360 шестидесятидесятиградусів
5 6400 міл
6 Метри, остання цифра - 0.1мм
7 Фути, остання цифра - 1/10 000фт
8 Метри, остання цифра - 0.01мм

Індекси полів формату GSI
Таблиця 2.2

Індекс Значення
11 Номер точки
12 Серійний номер інструмента
13 Тип інструмента
16 Номер станції
17 Дата
19 Час
21 Горизонтальний кут
22 Вертикальний кут
31 Похила відстань
32 Горизонтальна відстань
33 Різниця висот
41 Введений номер станції
42 Введене ім'я станції
43 Введена висота станції
51 Константи: ppm/константа призми
52 Число вимірів
53 Середньоквадратична похибка
58 Константа призми
59 ppm
81 Координата X цілі
82 Координата Y цілі
83 Висота цілі
84 Координата X станції
85 Координата Y станції
86 Висота станції
87 Висота відбивача
88 Висота інструмента

2.2. Структура форматного файлу GTS-6

Формат даних, з яким працюють прилади Торсон - це GTS-6 і GTS-7. У свою чергу, формат GTS-6 може бути відформатований і невідформатований. Файли, отримані з приладів Торсон в форматі GTS-6 і GTS-7, мають розширення TXT. Невідформатовані файли GTS-6 мають розширення FB0.

Основною одиницею файлу є запис, тобто рядок файлу.

Запис являє собою набір полів, розділених комами. Типи і кількість полів у записі визначаються типом самого запису. Тип запису визначається ідентифікатором, з якого починається рядок. Ідентифікатор має до восьми символів, при цьому відсутні символи заповнюються пробілами.

В даній роботі наведено опис файлу, отриманого програмою RGS.

Типи записів та поля формату GTS-6

Таблиця 2.3.

Тип запису Зразок запису Шифр поля Значення поля Примітка

Запис одиниць вимірювання UNITS a,b UNITS Ідентифікатор запису одиниць вимірювання

a одиниці вимірювання віддалей: M - метри; F - фути.

b одиниці вимірювання кутів: D - градуси; G - гони

Запис станції STN Nst,i,Code STN Ідентифікатор запису станції

Nst Назва пункту

i Висота інструмента

Code Код пункту

Запис координат XYZ Y,X,H XYZ Ідентифікатор запису координат. Якщо XYZ існує, то повинно бути після STN

Y Східна координата

X Північна координата

H Висота

Тип запису Зразок запису Шифр поля Значення поля Примітка

Запис пункту наведення BKB Ntr,A,R BKB Ідентифікатор запису напрямку орієнтування Якщо BKB існує, то повинно бути після STN

Ntr Назва пункту орієнтування

A Азимут на пункт орієнтування (програмою RGS не враховується)

R Відлік по горизонтальному кругу на пункт орієнтування

Запис точки SS Nr,V,Code SS Ідентифікатор запису точки, пункту

Nr Назва пункту, точки спостереження

V Висота наведення

Code Код пункту, точки

Запис виміру на точці R Відлік по горизонтальному кругу HV, SD и HD повинні бути після SS

HV R,B B Вертикальний кут

SD R,B,D D Похила відстань

Рис. 2.3. Зразок файлу GTS-6

2.3. Структура форматного файлу RW5

Щоб отримати дані з робочого інструменту його необхідно підключити до персонального комп'ютера та запустити спеціальну програму для експорту даних в файл RW5. Самі дані можуть відрізнятися залежно від того, який прилад використовувався під час знімання (тахеометр, роботизований тахеометр чи GNSS приймач) та яка програма була використана для експорту. В даній роботі були розглянені дані GNSS приймачів та використані програми «Cube-A» і «SurvCE».

Структура файлу RW5 складається шапки, на блоків точкових вимірів. В шапці знаходяться загальні дані, що стосуються всіх вимірів. Вона може містити інформацію про фізичні метрики антени. Блоки точкових вимірів складаються кількох послідовних рядків. Тип інформації в блоках залежить від методики та характеру проведених вимірювань.

Типи вимірів та поля формату RW5

Таблиця 2.4

Тип запису	Опис запису	Назва поля	Опис поля	Примітка
JB	Загальні дані робочої сесії	NM	Назва робочої сесії	
DT	Дата			
TM	Час			
MO	Запис налаштувань	AD	Азимут	
UN	Одиниці вимірювання віддалі.	0 - Фути, 1 - Метри, 2 - Американські фути		
SF	Масштаб			
EC	Кривизна Землі.	0 - неврахована, 1 - врахована		
EO	Стала призми			
AU	Кутові одиниці вимірювання			
LS	Проміжні висоти	HI	Висота інструмента	
HR	Висота віхи			
Antenna Type	Дані про антену	RA	Радіус антени	Не всі програми видають дану інформацію
SH	Нахил фазового центру			
L1	Зміщення фазового центру антени по висоті			
BP	Запис базової точки	PN	Номер точки	
LA	Широта			
LN	Довгота			
EL	Висота			
AG	Висота антени			
PA	Відстань від фазового центру до антени			
AT	Тип підключення з ПК			
SR	Характер мережевого з'єднання			
HR	Висота віхи введена користувачем			
GPS	Запис на точці	PN	Назва точки	
LA	Широта точки (WGS84)			
LN	Довгота точки (WGS84)			
EL	Висота над еліпсоїдом точки завжди в метрах			
GS	Запис редукованих локальних координат	PN	Номер точки	
E	Східна координата			
N	Північна координата			
EL	Висота			
GT	Місцевий час	PN	Номер точки	
SW	GPS тиждень на старті виміру			
EW	GPS тиждень в кінці виміру			
ST	Час в секундах від початку GPS тижня в кінці виміру			

Тип запису	Опис запису	Назва поля	Опис поля	Примітка
ET	Час в секундах від початку GPS тижня на старті виміру			
HSDV	Додаткові дані супутникових вимірів		Горизонтальне стандартне відхилення	
VSDV	Вертикальне стандартне відхилення			
STATUS	Статус розв'язку			
SATS	Кількість доступних супутників			
AGE	Час затримки сигналу			
PDOP	Погіршення точності через позиційну складову			
HDOP	Погіршення точності в горизонтальній площині			
VDOP	Погіршення точності у вертикальній площині			
TDOP	Погіршення точності через часову складову			
GDOP	Сумарне геометричне погіршення точності з часовою і позиційною складовою			
NSDV	Північне стандартне відхилення			
ESDV	Східне стандартне відхилення			

Рис. 2.4. Зразок файлу RW-5

В форматі також зазначено, якою програмою був експортований даний файл з приладу та багато іншої додаткової інформації, що може виявитися корисною для наукових та дослідницьких цілей.

2.4. Основи веб-програмування

Інтернет протоколи

Інтернет працює через мережу маршрутизації пакетів відповідно до протоколу Internet (IP), протоколу

управління транспортом (TCP) та інших протоколів.

Протокол - це набір правил, що визначають, як комп'ютери повинні спілкуватися один з одним в мережі. Наприклад, TCP протокол має правило, що якщо один комп'ютер надсилає дані на інший комп'ютер, то цільовий комп'ютер повинен повідомити вихідному комп'ютеру про відсутність будь-яких даних, щоб вихідний комп'ютер міг їх повторно надіслати. Або IP протокол, який визначає, як комп'ютери повинні спрямовувати інформацію до інших комп'ютерів, приєднуючи адреси до даних, які він надсилає.

Дані, що надсилаються через Інтернет, називаються повідомленнями. Перш ніж надсилати повідомлення, воно спочатку розбивається на багато фрагментів, званих пакетами. Ці пакети надсилаються незалежно один від одного. Типовий максимальний розмір пакета - від 1000 до 3000 символів. Інтернет-протокол визначає, як повідомлення повинні бути пакетовані.

Мережа маршрутизації пакетів - це мережа, яка здійснює маршрутизацію пакетів від вихідного комп'ютера до цільового комп'ютера. Інтернет складається з масивної мережі спеціалізованих комп'ютерів під назвою маршрутизатори. Завдання кожного маршрутизатора полягає в тому, щоб знати, як переміщувати пакети разом з їх джерелом до місця призначення. Під час подорожі пакет переміститься через декілька маршрутизаторів.

Коли пакет переходить від одного маршрутизатора до іншого, це називається хоп.

Клієнт-серверна модель

Модель "клієнт-сервер" - це розподілена програмна структура, яка розбиває завдання чи робоче навантаження між постачальниками ресурсу або послуги, що називаються серверами, і запитувачами послуг, що називаються клієнтами. У архітектурі клієнт-сервер, коли клієнтський комп'ютер надсилає запит на дані сервера через Інтернет, сервер приймає запит і доставляє запитувані пакети даних назад клієнту. Клієнти не діляться жодними своїми ресурсами. Прикладами моделі клієнт-сервер є електронна пошта, всесвітня павутина тощо.

1. Клієнт: Коли ми говоримо слово «Клієнт», це означає говорити про людину чи організацію, яка використовує певну послугу. Аналогічно в цифровому світі Клієнт - це комп'ютер (Хост), тобто той, хто здатний отримувати інформацію або використовувати певну послугу від постачальника послуг (Сервер).
2. Сервери: Аналогічно, коли ми говоримо слово "Сервери", це означає людину або середовище, яке щось обслуговує. Аналогічно в цьому цифровому світі Сервер - це віддалений комп'ютер, який забезпечує інформацією (даними) або доступ до певних послуг.

Рис. 2.5. Ілюстрація моделі клієнт-сервер

Отже, Клієнт в основному запитує щось, а Сервер обслуговує його доти, доки він присутній у базі даних.

Є кілька кроків, які слід виконати для взаємодії сервера і клієнта.

1. Користувач вводить URL-адресу (Uniform Resource Locator) веб-сайту чи файлу. Потім браузер запитує сервер DNS (Domain Name System).
2. Пошук DNS-сервера для адреси WEB-сервера.
3. DNS-сервер відповідає IP-адресою WEB-сервера.
4. Браузер надсилає запит HTTP / HTTPS на IP-адресу WEB-сервера (надається сервером DNS).
5. Сервер надсилає необхідні файли веб-сайту.
6. Потім браузер надає файли, і веб-сайт відображається. Ця візуалізація проводиться за допомогою інтерпретатора DOM (Document Object Model), інтерпретатора CSS та JS Engine, спільно відомого як компілятори JIT або (Just in Time).

Рис. 2.6. Ілюстрація взаємодії сервера і клієнта

DOM, Virtual DOM, Incremental DOM

DOM (аббревіатура від Document Object Model) - спосіб подання структурного документа за допомогою об'єктів. Це кросплатформенна і мовно-незалежна домовленість для представлення та взаємодії з даними в HTML, XML.

Веб-браузери обробляють складові DOM, і ми можемо взаємодіяти з ними, використовуючи JavaScript і CSS. Ми здатні працювати з вузлами документа, змінювати їхні дані, видаляти і вставляти нові вузли.

Рис. 2.7. DOM дерево

Головна проблема DOM - він ніколи не був розрахований для динамічного користувацького інтерфейсу (UI). Ми можемо працювати з ним, використовуючи JavaScript і бібліотеки на зразок jQuery, але їх використання не вирішує проблем з продуктивністю.

Після невеликого скролінгу в сучасних соціальних мережах, таких як Twitter чи Facebook, ми будемо мати десятки тисяч DOM-вузлів, ефективно взаємодіяти з якими - завдання не з простих.

Для прикладу, переміщення 1000 div-блоків на 5 пікселів вліво може зайняти більше секунди - це занадто багато для сучасного інтернету. Можна оптимізувати скрипт і використовувати деякі прийоми, але в підсумку це викличе лише головний біль при роботі з величезними сторінками і динамічним UI. Використання підходу з Virtual DOM вирішує цю проблему.

Virtual DOM не є стандартом і в кінцевому підсумку ми як і раніше взаємодіємо з DOM, але робимо це якомога рідше і більш ефективно.

Замість того, щоб взаємодіяти з DOM безпосередньо, ми працюємо з його легкою копією. Ми можемо вносити зміни в копію, виходячи з наших потреб, а після цього застосовувати зміни до реального DOM. При цьому відбувається порівняння DOM-дерева з його віртуальною копією, визначається різниця і запускається «перемальовування» того, що було змінено.

Такий підхід працює швидше, бо не включає в себе всі великовагові частини реального DOM.

Рис. 2.8. Розпізнавання змін у Virtual DOM

Однак нещодавно компанія Google знайшла ще ефективніше рішення для роботи з DOM - Incremental DOM реалізоване в JavaScript фреймворку Angular. Incremental DOM не вимагає пам'яті для повторного рендерингу представлення, якщо воно не вносить зміни в DOM. Пам'ять необхідно буде виділити тільки

в тому випадку, якщо будуть додані або видалені DOM-вузли, а обсяг виділеної пам'яті буде пропорційний виробленим змін в DOM.

2.5. Етапи створення веб додатку

Створення веб додатку зазвичай передбачає наступні етапи:

1. Розуміння потреб клієнта
2. Глибокі дослідження та аналіз
3. Планування
4. Дизайн
5. Розробка
6. Тестування та розгортання
7. Обслуговування

1. Розуміння потреб клієнта. Досить часто вважають, що процес розробки веб-сторінок починається з розробки та розробки, але факт полягає в тому, що ці етапи приходять досить пізно. Перший крок і справді найважливіший (і часто його ігнорується) - розуміння потреб клієнта.

Виявлення та розуміння того, чого саме хоче клієнт допомагає забезпечити ідеальне рішення, яке він шукає. У деяких випадках, коли клієнт має технічну освіту, багато простіше зрозуміти потреби та технології, які він хоче бачити в проекті. Однак, коли клієнт повністю невідомий програмування, треба задати додаткові питання та роз'яснення, що допоможуть краще осягнути мотивацію і проблематику клієнта, знайти оптимальне ринкове рішення.

2. Глибокі дослідження та аналіз. Кожна програма відрізняється одна від одної. Тож на цьому етапі команда досліджує та збирає якомога більше релевантної інформації для проекту. Веб-додаток для електронної комерції, що продає чоловічий одяг, відрізнятиметься від інтернет-ринку роботи. Таким чином, глибоке дослідження та аналіз галузі, цільової аудиторії, конкурентів, девізу проекту тощо надають розуміння та знання, необхідні для розробки бажаного веб-додатку.

3. Планування. Як тільки команда отримує ведучі шаблони та мапу додатку, вона переходить до створення кожної сторінки веб-програми. На цьому етапі графічні дизайнери подають життя затвердженим каркасам із власною графікою, логотипом, кольорами, типографією, анімацією, кнопками, випадаючими меню та іншим на основі потреб проекту. Дизайн програми має вирішальне значення для досвіду користувача. Перше враження, яке користувачі мають на веб-сайті, становить аж 94%, що стосується дизайну. Отже, обов'язково потрібно переконатися, що веб-додаток естетично приваблює цільову аудиторію. Навіть найменші деталі, такі як тіні графіки або колір кнопки заклик до дії, повинні бути точно враховані. Насправді кольори веб-сайту відіграють монументальну роль у забезпеченні кращого досвіду для користувачів. Згідно з дослідженням, споживачі формують початкове оцінку продукту протягом 90 секунд взаємодії, і 62% -90% їх базується на кольорі. Крім того, різні кольори можуть викликати різні емоції.

5. Розробка. Це етап, коли проекти, затверджені клієнтом, перетворюються на робочу модель. Процес розробки можна розділити на дві частини, тобто фронтенд і бекенд. Frontend Development - це розробка клієнтської програми, яку бачать користувачі. Усі конструкції, зроблені на попередньому етапі, перетворюються на HTML-сторінки з необхідними анімаціями та ефектами. І для додання деяких складних функціональних можливостей використовуються JavaScript фреймворки чи бібліотеки, такі як Angular, React, Vue тощо. Враховуються потреби мобільних пристроїв.

Backend Development - це розробка програми на стороні сервера, яка є «душею» фронту і перетворює користувальницький інтерфейс у робочий веб-додаток. Бекенд розробники створюють серверний додаток, базу даних, інтегрують бізнес-логіку і все, що працює «під капотом».

6. Тестування та розгортання. Після того, як веб-додаток створений, перед його розгортанням на сервері, він проходить кілька ретельних тестів, щоб переконатися у відсутності помилок та проблем. Команда контролю якості проводить такі тести, як тест на функціональність, тест на зручність, тест на сумісність, тест на ефективність тощо. Це робить веб-додаток готовим до роботи, запуску. Крім того, тестування також допомагає відкрити шляхи вдосконалення веб-програми найближчим часом. Після того, як команда із забезпечення якості «показує зелений прапор» веб-програма розгортається на сервері за допомогою FTP (протокол передачі файлів).

7. Обслуговування. Процес веб-розробки не закінчується після розгортання. Існує кілька завдань після розгортання, які повинна виконувати компанія з веб-розробки, такі як надання клієнтам вихідного коду та проектних документів, робота над їх відгуками та підтримка після розгортання. Цей етап має однакоку ступінь важкості, оскільки реальна мета веб-додатка починається, як тільки він з'являється для користувачів. Подальші зміни відповідно до відгуків, підтримка та обслуговування користувачів однаково необхідні.

РОЗДІЛ 3. РОЗРОБКА ТА ДОСЛІДЖЕННЯ РОБОТИ ОНЛАЙН КОНВЕРТОРА

3.1 Створення конвертера

Більшості сучасних конвертерів має наступні недоліки:

1. Залежність від типу та версії операційної системи, платформи. Така залежність створює необхідність в постійній підтримці. Адаже за відсутності постійної чи ситуативної робочої групи, яка контролюватиме стабільність і якість роботи програмний продукт швидко втратить актуальність, застаріє і створюватиме загрозу для персональних даних на пристрої користувача.

2. Стаціонарність консольних програм постійно створює необхідність їх повторного встановлення в операційній системі при зміні девайсу.

3. Некоректність чи незрозуміла логіка обрахунків даних

4. Цінова недоступність для навчальних чи наукових цілей.

Квінтесенцією цієї магістерської кваліфікаційної роботи було створити самостійний, кросплатформенний, кросбраузерний, інтуїтивний, корисний та доступний продукт, що вирішить проблему конвертування вихідних форматних файлів та приведе їх до зручного вигляду.

В зв'язку з цим було прийнято ряд архітектурних рішень. Серед основних - використання вебу, як універсальної платформи для функціонування додатку. Мовою вебу, де факто, є динамічно-типизована, об'єктно-орієнтована мова програмування JavaScript. Однак, для безпечнішої і швидшої розробки було обрано TypeScript - розробку компанії Microsoft. TypeScript транспілюється в JavaScript. При цьому, він надає можливість контролювати тип вхідних на вихідних параметрів функції через вбудовані тими і користувацькі інтерфейси.

В ході розробки будь-якого великого проекту рано чи пізно розробники створюють власні інструменти,

бібліотеки, методології, які полегшують розробку. Таким чином, була створена найпопулярніша JavaScript бібліотека для розробки SPA (Single Page Applications) - «React.js», яку використовує та розвиває Facebook.

Для онлайн конвертера було використано альтернативне середовище розробки з відкритим програмним кодом, пропонуване компанією Google - Angular Framework. Чому Angular? Він підтримується на різних платформах (веб, мобільні платформи, консолі), він потужний, сучасний, має приємну екосистему. Angular представляє не тільки інструменти, але й шаблони дизайну, щоб побудувати гнучкий і легкопідтримуваний проект. Якщо програма на Angular створена належним чином, то її код зручно структурований, і тому не потрібно буде витрачати багато часу, щоб зрозуміти, що в ній відбувається. У Angular весь проект розбитий на окремі структурні блоки - компоненти. В свою чергу компоненти чітко розділені на HTML шаблон, CSS стилі та TypeScript клас, що керує даними. Окремо можна додати файли для тестування, маршрутизації, керуючого модуля.

Рис. 3.1. Зразок структури компоненти «Not Found» в навігаційному блоці текстового редактора «Visual Studio Code».

Програма «FormatStandardify» не має серверної частини, оскільки не потребує її. Всі обрахунки здійснюються на стороні клієнта, тобто в браузері.

Наступна ілюстрація показує схему роботи програми:

Рис. 3.2. Схема роботи програми

Частина конвертера, що парсить текстовий файл ділиться на декілька етапів: завантаження, первинний парсинг, вторинний парсинг.

Етап завантаження відповідає за перетворення текстового файлу в стрічку, як структуру даних JavaScript. Потім ця стрічка передається в компаратор формату, де залежно від обраного користувачем формату викликається відповідний сервіс.

Рис. 3.3. Компаратор вхідного формату

В сервісі починається первинний парсинг, який полягає в групуванні точок.

Кожен сервіс розбиває вхідну стрічку на рядки, відділяє метадані від даних вимірів. Там, де це необхідно, ділить дані по станціях стояння чи базових станціях (далі просто станціях), а кожній в станції виокремлює конкретні точки. Після цього, всі дані й досі залишаються в стрічках, тобто звичайних текстових рядках, але поділені на станції.

Рис. 3.4. Результат роботи первинного парсингу.

Перші три стрічки, обведені червоними прямокутниками - це шапка файлу. В першій стрічці закодовані дані про назву файлу, дату та час створення його файлу. В другій стрічці закодовані, одиниці вимірювання кутів та віддалей, масштаб, стала призми і факт врахування чи неврахування кривини Землі.

Починаючи з 7 стрічки і до кінця файлу знаходяться самі дані вимірів. На рисунку 3.1.4 це один червоний блок, який позначає одну станцію. Однак таких блоків може бути декілька. В середині блоку станції знаходяться два зелені блоки. Перший блок містить дані, що стосуються станції: назва станції, широта, довгота і висота станції, азимутальний напрям, висота антени та висота віхи. Другий блок містить точки, межі котрих виокремлені синіми прямокутниками.

Після того, як точки згруповані, починається вторинний парсинг. На цьому етапі масив стрічок перетворюється в об'єкт з властивостями. Для цього була створена функція-компаратор для полів стрічки даних.

Рис. 3.5. Функція-компаратор полів стрічки даних.

Ця функція розщеплює стрічку на поля. В даному випадку сепаратором виступає кома «,» (рис. 3.1.4). Після цього, відповідно до заголовка, функція розпізнає тип значення та записує його в об'єкт точки.

Рис. 3.6. GNSS вимір перед парсингом

Рис. 3.7. GNSS вимір після парсингу, виведений в консолі розробника у браузері Google Chrome

Останнім етапом парсингу є розгортка об'єктів JavaScript в електронні таблиці EXCEL. Для цього в дерево залежностей додатку було додано бібліотеки «sheetjs» та «xlsx». Функції цих бібліотек приймають двовимірний масив даних, де перший рядок являється заголовком таблиці, а всі наступні рядки містять дані точок.

На виході отримуємо дані в електронних таблицях EXCEL.

Рис. 3.8. Дані GNSS вимірів після конвертації в електронні таблиці EXCEL

Для зручності роботи з даними, всі лінійні виміри автоматично приводяться до метрів, а кутові виміри до десяткових градусів і додатково вказуються у заголовках полів. Для обрахунків JavaScript використовує стандарт «float64». Даний стандарт має точність - 53 біти, або 16 знаків після десяткової коми.

Рис. 3.9. Вкладка браузера із робочим конвертером

Вихідний код веб додатку зберігається в публічній репозиторії «FormatStandardify» користувача під нікнеймом «KeinOhrHasen» на платформі GitHub». Це сервіс був обраний через те, що він надає можливість не тільки керувати версійністю програми, а й забезпечує її хостинг за умови відсутності серверної частини.

Посилання на репозиторій: <https://github.com/KeinOhrHasen/FormatStandardify>

Посилання на працюючий конвертер: <https://keinohrhasen.github.io/FormatStandardify/converter>

Рис. 3.10. Сторінка репозиторію конвертера

На момент захисту магістерської кваліфікаційної роботи даний сервіс має 14 активних користувачів.

3.2 Дослідження конвертованих даних з формату RW5

Для перевірки правильності формування звіту в електронних таблицях EXCEL було проведено

дослідження на 4 пунктах лінійного базису лабораторії GEOTerrace.

Рис. 3.11. Контрольні виміри на території лабораторії GEOTerrace

Для цього були проведені вимірювання GNSS приймачем фірми Stonex на 5, 7, 10, 15 пунктах. Після цього дані з приладу були експортовані в формат RW5 та конвертовані в таблиці EXCEL тестовою версією конвертера.

Рис. 3.12. Мережа GEOTerrace на гугл-картах із заданими координатами

Рис. 3.13. Порівняння місцевих координат базису (текстовий файл pblh) і геодезичних координат (текстовий файл pblh) базису із конвертованими даними з приладу Stonex.

Як видно з рисунка 3.1.3 формат rw5 враховує висоту антени як суму висоти віхи і висоти фазового центру антени для місцевих координат. Однак ігнорує висоту антени для геодезичних та прямокутних координат. Така поведінка цілком неочевидна. Тому користувач може і не підозрювати про таку логіку. Задля зручності представлення всіх координат в єдиній точці було прийнято рішення віднімати висоту антени від геодезичних координат та згідно нових висот обраховувати прямокутні координати.

Рис. 3.14. Порівняння місцевих координат базису (текстовий файл pblh) і геодезичних координат (текстовий файл pblh) базису із конвертованими даними з приладу Stonex при врахуванні висоти антени для геодезичних та прямокутних координат

ВИСНОВКИ

У даній магістерській кваліфікаційній роботі була порушена проблема конвертації вихідних форматів геодезичних приладів для забезпечення уніфікації вирішення наукових та прикладних задач. Ця проблема стосується як метрологів і науковців, так і користувачів, які мають необхідність у формуванні звітів польових робіт методом GNSS у RTK режимі, контролі якості виконаних робіт.

Для її вирішення були проведені:

1. Аналіз наявних програмних рішень («Редактор Измерений», «RTK Format 4.0.0», «SurvCE RW5 Report»), на основі якого, були укладені вимоги до конвертера.
2. Детальне дослідження текстових форматів gsi, rts-5, rw5 для подальшої конвертації в електронні таблиці EXCEL.
3. Опис методів комп'ютерного програмування та веб-програмування.
4. Розробка веб-додатку для конвертації форматних файлів.
5. Запуск білда проекту на хостинговому сервісі «GitHub Pages»
6. Складання кошторису виконаних робіт.

Результатом виконаної роботи є працюючий онлайн-конвертер, що задовільняє потреби більше десятка активних користувачів.