

Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Дипломна робота
магістра

з теми «**ВЕБ-ОРІЄНТОВАНА СИСТЕМА КЕРУВАННЯ
ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ**»

Виконав: здобувач вищої освіти
2 курсу, групи KN1-M22
спеціальності 122 Комп'ютерні науки
Нарольський Денис В'ячеславович

Керівник: **Федорчук В. А.**,
професор кафедри комп'ютерних наук,
доктор технічних наук, професор

Рецензент: **Ковальська І. Б.**,
доцент кафедри математики, кандидат
фізико-математичних наук, доцент,

Кам'янець-Подільський – 2023

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ОСОБЛИВОСТІ ПРОЦЕСІВ КОМП'ЮТЕРИЗАЦІЇ В ГАЛУЗІ ПОЛІГРАФІЧНОЇ ПРОМИСЛОВОСТІ.....	5
1.1. Історія розвитку	5
1.2. Поліграфічна промисловість України	11
1.3. Перспективи галузі	15
РОЗДІЛ 2. ОГЛЯД ПРОБЛЕМАТИКИ КЕРУВАННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ У ВЕБ-ОРІЄНТОВАНИХ СИСТЕМАХ	16
2.1. Аналіз існуючих підходів до керування технологічним процесом	21
2.2. Огляд баз даних, зокрема SheetDB, для збереження та обробки формул	28
2.2.1. Кроки для початку користування SheetDB:	30
2.3. Структура та архітектура веб-системи для керування технологічним процесом.....	31
2.3.1. Компоненти системи.....	31
РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ ТА ТЕСТУВАННЯ.....	35
3.1. Розробка інтеграції Excel формул до бази даних SheetDB	35
3.2. Програмування та реалізація веб-інтерфейсу для роботи з цими даними	40
ВИСНОВКИ	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45

ВСТУП

Розвиток веб-технологій розширює горизонти можливостей для різних галузей промисловості та підприємств. У сучасній динамічній реальності комп'ютерні технології стають важливим інструментом у забезпеченні ефективності та оптимізації робочих процесів, особливо в такій галузі, як поліграфія.

Поліграфічна справа, як складова сучасного ринку, вимагає не лише високої якості продукції, а й швидкості реакції на змінні вимоги замовників. Розуміння важливості оптимізації технологічних процесів та використання веб-орієнтованих інструментів у цій галузі стає ключовим фактором конкурентоспроможності.

Саме тут виникає необхідність у створенні інформаційної системи, що спрямована на керування технологічними процесами поліграфічного виробництва. Інтеграція технологій веб та використання високоефективних інструментів, таких як Excel і SheetDB, відкриває можливості для оптимізації обчислювальних процесів прямо на сайті, що спрощує та прискорює робочі завдання.

Дослідження та розробка такої web-орієнтованої системи керування технологічним процесом для поліграфічного підприємства не лише дозволить покращити ефективність виробництва, але й стане кроком уперед у використанні передових технологій у цій сфері.

Завдяки стрімкому розвитку веб-технологій, використання SheetDB, MongoDB та React надає можливість створювати веб-додатки, які поєднують у собі ефективність, швидкість розробки та високу функціональність. Розробка таких застосунків стає ключовою у контексті сучасних вимог до користувацького досвіду та технічної ефективності.

Ця дипломна робота спрямована на розгляд важливості і можливостей, які відкриваються завдяки використанню web-технологій у поліграфічній галузі, а також на розробку та практичне впровадження системи, що

спрямована на підвищення ефективності технологічних процесів на підприємстві.

Мета: Розробка веб-застосунку на основі SheetDB, MongoDB та React для створення інтерактивних веб-сайтів з високою продуктивністю і зручним інтерфейсом користувача.

Об’єкт дослідження. Процеси створення сучасних веб-застосунків на основі розробки інтерактивних та високопродуктивних веб-сайтів.

Предмет дослідження. Використання SheetDB, MongoDB та React у створенні сучасних веб-застосунків.

Для досягнення поставленої мети сформульовано **завдання дослідження**, а саме:

1. Аналіз предметної області.
2. Огляд проблематики керування технологічним процесом у веб-орієнтованих системах.
3. Аналіз існуючих підходів до керування технологічним процесом.
4. Огляд баз даних, зокрема SheetDB, для збереження та обробки формул.
5. Розробка структури та архітектури веб-системи для керування технологічним процесом.
6. Розробка інтеграції Excel формул до бази даних SheetDB.
7. Програмування та реалізація веб-інтерфейсу для роботи з даними.

Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел.

РОЗДІЛ 1. ОСОБЛИВОСТІ ПРОЦЕСІВ КОМП'ЮТЕРИЗАЦІЇ В ГАЛУЗІ ПОЛІГРАФІЧНОЇ ПРОМИСЛОВОСТІ

1.1. Історія розвитку

Поліграфія пройшла тривалий і складний шлях розвитку. Її технічною основою є винайдення близько 1440 Й. Гутенбергом книгодрукування. Вже в XVI ст. поліграфія набула характеру розвиненої мануфактури. У XIX ст. з винаходом друкарської машини в поліграфії відбувається промислова революція, що ознаменована створенням поліграфічного машинобудування. У середині XX ст. у ході науково-технічної революції поліграфія розвивається за такими напрямками:

- перехід до електронних способів виготовлення друкарських форм для всіх способів друку (використання комп'ютера для кольороподілу та фотонабору),
- широке застосування рольового офсетного друку на високошвидкісних машинах,
- створення автоматичних потокових ліній в оздоблювальних цехах,
- комплексна механізація й автоматизація всього виробництва,
- використання локальних та глобальних комп'ютерних мереж для передачі даних.

Технологічні процеси

Технологія поліграфії містить три основні групи виробничих процесів:

- додрукарські (підготовчі),
- друкарські (власне сам процес друку),
- післядрукарські (оздоблювальні).

Додрукарські процеси спрямовані на виготовлення оригінал-макету, друкарської форми. Завдання друкарських процесів — отримання тиражних

відбитків, що відтворюють оригінал. Оздоблювальні процеси завершують виготовлення друкованої продукції.

Додрукарські процеси включають усі процеси до стадії виготовлення друкарських форм. За допомогою комп'ютерів виготовляють оригінал-макет видання, якій потім переносять на друкарську форму. Текстова форма може бути отримана з типографського шрифтового методу ручного набору або ж за допомогою набірних машин (застарілий метод, зараз не використовується), комплектуючі літери й виготовляючи форму або її напівфабрикат (у вигляді, наприклад, фотокопії). Широке поширення одержали різні методи автоматизованого набору. Ручні способи виготовлення ілюстрованої друкарської форми в сучасній поліграфії використовуються як методи станкової графіки або для створення оригіналів, згодом відтворених фотомеханічним або іншим шляхом.

Серед ручних способів високого друку найбільшою популярністю користуються:

- гравюра на дереві, або ксилографія,
- ліногравюра;

глибокого друку:

- різцева гравюра на металі,
- офорт,
- акватинта;

плоского друку:

- літографія.

Ілюстраційні форми виготовляють фотомеханічними методами (автотипія, фотоцинкографія, фотолітографія, фототипія та ін.), а також за допомогою електронних гравіювальних машин. Оригінал, що підлягає

фотомеханічному репродукуванню, фотографують на репродукційному фотографічному апараті (напівтонові оригінали фотографують через растр). Негативне або позитивне зображення в копіювальних рамах копіюють на металеву пластину, на яку попередньо нанесений світлочутливий шар. Потім пластину піддають відповідній обробці й травленню в машинах для травлення. При відтворенні багатоколірних оригіналів виготовляють кольороподільні друкарські форми за допомогою кольороділильного фотографування або із застосуванням електронних кольороподільовачів — колірокоректорів.

Розрізняють 3 види друкарських процесів:

- друкування з формуванням зображення в фарбовому резервуарі (гектографія, туркинотипія);
- друкування з формуванням зображення на проміжній поверхні — формі (так звані класичні способи друку);
- друкування з формуванням зображення на сприймаючій поверхні (способи з електростатичним й електромагнітним переносом фарбового шару).

Класичні способи друку розрізняються залежно від методу поділу друкуючих і пробільних елементів. Формування фарбового шару може здійснюватися як у процесі переміщення фарби крізь форму (трафаретний друк, ротатор), так і шляхом нанесення фарби на поверхню форми. В останньому випадку застосовується просторова (високий і глибокий друк) або фізико-хімічна (плоский друк) поділ друкарських і пробільних елементів. У друкарських процесах, що використовуються в сучасній поліграфії фарбове зображення переносять із форми на сприймаючу поверхню безпосередньо або ж за допомогою однієї (офсетний друк) або двох (орловський друк) проміжних поверхонь. Друкування здійснюється на друкарських машинах, які розрізняються за способом друку, за схемою побудови друкарського

пристрою, за кількістю переносів фарбового шару, за типовою подачею сприймаючих поверхонь. Перед друкуванням проводиться ряд підготовчих процесів: розміщення, або спуск смуг форми, її закріплення, приводка, приправка.

Сукупність формних і друкованих процесів й устаткування, що застосовується для них, призначених для відтворення малотиражної документації (звичайно інформаційного або управлінсько-адміністративного характеру) називається оперативною поліграфією.

Характер оздоблювальних процесів залежить від виду друкованої продукції. Найскладніші брошурувальні процеси, застосовувані в процесі виготовлення книг і журналів.

Поширення набуло ламінування обкладинок книг.

Поліграфічна промисловість

Поліграфічні підприємства залежно від характеру виробництва називаються друкарнями, типолітографіями, фабриками кольорового друку, офсетна фабрика й т.д.; підприємства, що поєднують кілька типографських процесів або видів друку, називаються поліграфічними комбінатами.

За видом друкарської продукції, що випускається, підприємства можуть бути універсальними або спеціалізованими (газетні, газетно-журнальні, книжкові, картографічні й т. д.).

Основні етапи технологічного процесу виготовлення друкованої продукції (додрукарська підготовка, друк, післядрукарська обробка) Виробництво друкованої продукції в більшості випадків складається з трьох або чотирьох роздільних, але взаємозалежних процесів:

- обробка текстової й образотворчої інформації – оригіналів, що підлягають поліграфічному відтворенню. У результаті виконання цього

процесу одержують негативи або діапозитиви на прозорій плівці, що містять інформацію друкарських форм;

- виготовлення з негативів або діапозитивів комплекту друкарських форм, необхідних для розмноження інформації;
- друкування тиражу – одержання з друкарських форм певної кількості ідентичних видрукованих аркушів, зошитів і т.ін. (власне розмноження інформації);
- виконання брошурувальних або брошурувально-палітурних робіт, оздоблювальних процесів.

На цій стадії продукція набуває зручний для використання інформації вид. Перші два процеси часто називають додрукарськими процесами, третій і четвертий можуть бути об'єднані та виконуватися як єдиний процес на спеціалізованому друкарському устаткуванні. В останні роки з розвитком обчислювальної техніки з'явилася можливість об'єднати в єдиний технологічний цикл усі чотири процеси. Сучасні цифрові друкарські машини та комплекси дозволяють в автоматичному режимі виконувати всі операції, починаючи з обробки оригіналу і закінчуючи готовою продукцією. До додрукарських процесів включають операції, спрямовані на обробку авторських або видавничих оригіналів з метою одержання в кінцевому результаті друкарської форми. Ці процеси є принципово однаковими як для великої, так і для малої поліграфії, але залежно від виду оригіналів, обраної схеми процесу обробки, вимог до якості результатів та ін. вони можуть бути більш чи менш розвиненими та ускладненими. Для сучасної поліграфії характерним є використання у додрукарських процесах комп'ютерних технологій, на цьому і побудована схема додрукарських процесів. Одним з основних етапів додрукарської підготовки видання є створення макета. Процес створення макета своєю чергою складається з взаємозалежних етапів:

- макетування (начерк макета, вибір формату й орієнтації сторінки, задання полів, розробка модульної сітки, підбору елементів дизайну);
- підготовка тексту (підбор основного тексту, заголовків, таблиць);
- підготовка ілюстрацій (сканування графічних зображень, редагування розмірів, яскравості, контрастності, кольоровості, усунення муару й інших дефектів, вибір формату);
- вибір шрифтів (гарнітури, кегля, накреслення);
- верстка видання (визначення довжини рядка, ширини стовпчика, способів вирівнювання, формування переносів, завдання міжлітерних, міжслівних і міжрядкових інтервалів);
- друк оригінал-макету (вибір принтера і параметрів друку, спуск шпальт, калібрування принтера).

Друкарські процеси розглянуто на прикладі підприємств, де властиве використання аркушевих друкарських машин, і подані схемою У друкарнях, де брошурувально-палітурні процеси направлені на виготовлення брошури чи книжки головним чином у м'яких обкладинках або на обробку видрукованих аркушів з метою оздоблення готової продукції, післядрукарські процеси частіше називають брошурувально-обробними та оздоблювальними процесами.

Перед здійсненням тиражного друкування необхідно перевірити якість проведених перетворень інформації шляхом синтезу кольорової репродукції методом пробного друку.

1.2. Поліграфічна промисловість України

Поліграфічна промисловість — галузь економіки, що об'єднує підприємства, які виготовляють друковану продукцію (книги, газети, журнали, плакати, географічні карти тощо), є матеріально-технічною базою для видавничої справи.

В Україні галузь посідає особливе місце в структурі національної економіки, її продукція призначена насамперед для задоволення освітніх, наукових, культурних та інформаційних потреб суспільства.

Характеристика

Доба незалежності відкрила для України широкі можливості інтегруватися у міжнародний інформаційний простір. У 1990-х на розвиток поліграфії вплинули криза книговидання України, зниження обсягу випуску книжкової продукції як за назвами, так і за накладом. Водночас зростала кількість періодичних видань, Україна друкувала власні гроші та цінні папери, поштові марки, конверти. Відкрився вільний доступ до світового ринку технологій та комп'ютерної техніки. Було прийнято закони про авторське право та видавничу справу. Значно покращувалася якість друкованої продукції за рахунок нових імпортованих матеріалів, впровадження у виробництво нових технологій.

Закінчилася монополія державної власності у підприємстві. Частина поліграфічних підприємств була приватизована й набула форми товариства, інші — залишилися у державній та комунальній власності. З'явилося багато невеликих поліграфічних структур, здебільшого приватної форми власності, зорієнтованих на випуск етикеткової та кольорової аркушевої продукції. У результаті конкуренції між виробниками поліграфічної продукції багато новостворених приватних підприємств виявилися нежиттєздатними.

У 2000-х роках з'явилися нові підприємства, створено мережу дистриб'юторських фірм, що сприяли впровадженню нових технологій, техніки, налагоджували сервісне обслуговування. Впроваджувалися комп'ютерні автоматизовані системи переробки текстової та графічної інформації (АСПТГІ), нові імпортні формні процеси та друкарське устаткування.

Кінець 20 — початок 21 ст. для поліграфічної галузі став етапом розвитку друкарства, на якому міцні позиції зайняли цифрові методи друку і комп'ютерні технології. У сучасному виробництві поєдналися висока технологічність, мобільність і складність процесів. Розвиток нових технологій, швидкі темпи впровадження їх у виробництво та досконалі витратні матеріали призвели до зростання конструкційної складності та художньо-технічної яскравості друкованих видань, появи нових видів продукції. Попри те, що накладі зменшуються, загальний обсяг і репертуар видань, досконалість конструкцій постійно зростають. Урізноманітнюються формати, збільшується колірність друкованої продукції, розширюються варіанти художньо-технічного оздоблення. Приватні, колективні поліграфічні підприємства, оснащені сучасною технікою для друку, успішно конкурують на ринку пакувальної, рекламної, бланкової та іншої комерційної продукції.

За останнє десятиліття розвиток поліграфічних підприємств значно змінився. Відбувається тотальна автоматизація та комп'ютеризація поліграфічного обладнання, поширення технологій з використанням фарб ультрафіолетового затвердіння і металізованих фарб, впровадження у виробництво гібридних технологічних процесів і устаткування. Зросла швидкість друку, зменшився час переналагодження та переходу з тиражу на тираж, максимально автоматизовано друкарський процес, електронний контроль якості кінцевої продукції, завдяки чому зводиться до мінімуму

використання трудових ресурсів. Спостерігається тенденція підвищення рівня якості поліграфічного виконання друкованої продукції.

Найпоширенішим способом друку є плоский офсетний. Він залишається основним способом відтворення різних видів друкарської продукції: газет, журналів, книг, художніх альбомів, етикеток, упакування, різноманітної акцидентної продукція. Серед новітніх тенденцій — загострення технологічної конкуренції між традиційним офсетним друком та цифровим друком.

Серед проблем розвитку поліграфічної галузі — відсутність іноземних інвестицій, брак обігових коштів, високі податки, низька рентабельність виробництва та продаж, відсутність державної підтримки, неврегульованість законодавчої бази, низька платоспроможність населення, залежність підприємств від імпорту матеріалів і устаткування тощо. Щороку на 5–10 % зменшується об'єм замовлень поліграфічної галузі. У найкращому становищі перебувають малі підприємства, невеликі друкарні, що спеціалізуються на випуску дрібних накладів. Вони отримують прибутки завдяки мобільності, невеликій кількості обладнання та малому штату працівників. Великі підприємства виживають передусім за рахунок виготовлення пакувальної, рекламної, бланкової та іншої комерційної продукції.

Якщо у 2007–2009 кількість поліграфічних підприємств зростала, то надалі зменшувалася у середньому на 10 % щорічно. 2012 в Україні функціонувало 2499 підприємств поліграфічної промисловості, з них переважна більшість — малі підприємства (91,3 %). Частка суб'єктів приватної форми власності — понад 88,9 %. У 2010–2013 динаміка обсягу інвестицій була позитивною, зокрема зросла на 19,76 % у 2010–2011. Але від 2013 обсяг інвестицій у поліграфічну галузь знижується щороку на 0,2–0,5 %.

За офіційними даними 2016, частка поліграфічних підприємств становила тільки 4,77 % від загальної кількості підприємств переробної

промисловості. За 2010–2016 їх кількість скоротилась на 268 одиниць (на 15 %); чисельність працівників, зайнятих у галузі — з 20,7 тис. до 18,8 тис. осіб (на 9,2 %). Причинами такої тенденції стали збитковість бізнесу, нестабільний фінансово-економічний стан підприємств. У 2015–2018 чистий прибуток отримували тільки 75–79 % виробників поліграфічних послуг.

Станом на перше півріччя 2019 у регіональній структурі поліграфічної галузі домінують: м. Київ — 2920 видавничо-поліграфічних підприємств, Харківська обл. — 883, Дніпропетровська — 393, Донецька — 382, Львівська — 341, Київська — 239, Одеська — 215. Диспропорція в розміщенні підприємств у регіонах призводить до деформації структури промисловості, зменшення рівня задоволення потреб споживачів у друкованій продукції.

Обсяг реалізованої продукції у вартісному вимірі упродовж 2010–2016 зростав нерівномірно, а 2017 значно знизився. Кількість друкованої продукції, за винятком рекламної, за цей період стабільно скорочувалась. Зниження обсягів реалізації відбувалось найбільшими темпами за періодичними виданнями. Єдиним видом друкованої продукції, обсяги якої зростають високими темпами, є виготовлення рекламної продукції (проспекти, плакати) — 31,2 % у середньому щорічно. Такі тенденції в поліграфічній промисловості спричинені загальним зниженням інтересу суспільства до читання, спадом промисловості та недостатньою державною підтримкою галузі, низькою конкурентоспроможністю вітчизняної продукції, зменшенням реальних доходів населення, а також появою інноваційних видів друку, переходом друкованих ЗМІ в он-лайн формат тощо. Поліграфічна галузь постала і перед глобальними викликами, що зумовлені зростанням попиту на цифрові технології та переходом від друкованих продуктів до цифрових носіїв.

Після проголошення незалежності України в поліграфії утворена нова підгалузь — виготовлення грошових знаків, цінних паперів, документів

суворої звітності. У короткі терміни для неї створено потужності, оснащені сучасною технікою, підготовлено висококваліфіковані кадри. Завдяки цьому держава змогла забезпечити себе цією стратегічно важливою продукцією. Водночас дії Міністерства фінансів щодо розширення переліку документів суворої звітності призвели до необґрунтованого монополізму та значного подорожчання такої продукції.

Серед нових тенденцій розвитку поліграфії та ринку поліграфічних послуг в Україні:

- орієнтація на екологічно-вартісні технології та захист довкілля;
- поширення цифрового друку;
- зростання інтересу до нестандартної поліграфії (ламінування, конгревне тиснення, биговка тощо);
- підвищення попиту на друк на пакуванні;
- впровадження нових маркетингових і бізнесових технологій тощо.

1.3. Перспективи галузі

Необхідною умовою прискорення розвитку галузі є державне сприяння та підтримка, захист інтересів підприємств, діяльність яких пов'язана з поліграфічним виробництвом та його обслуговуванням у державних, недержавних і громадських органах та організаціях. Важливим є ухвалення законодавчих актів і нормативних документів з метою поліпшення економічних і правових умов діяльності вітчизняних виробників поліграфічної продукції. Поліграфічна галузь України удосконалюватиме виробництво продукції завдяки впровадженню новітніх комп'ютерних технологій, апаратно-програмних комплексів, автоматизації всіх ланок видавничо-поліграфічного комплексу, задовольняючи потреби суспільства у доступі до інформації.

РОЗДІЛ 2. ОГЛЯД ПРОБЛЕМАТИКИ КЕРУВАННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ У ВЕБ-ОРІЄНТОВАНИХ СИСТЕМАХ

У веб-орієнтованих системах управління технологічним процесом виникають деякі основні виклики:

1. **Інтеграція технологій:** Управління технологічними процесами у веб-системах часто потребує інтеграції різноманітних технологій, платформ та джерел даних. Проблеми можуть виникати при спробі забезпечити сумісність і взаємодію між різними компонентами системи.
2. **Безпека даних:** Збільшена вразливість веб-систем до кібератак потребує посиленої уваги до захисту даних, особливо при управлінні технологічними процесами, де можуть бути конфіденційні дані про виробництво, матеріали тощо.
3. **Складність масштабування та продуктивності:** При обробці великої кількості даних у реальному часі, системи можуть стикатися з проблемами продуктивності та масштабованості. Ефективне управління технологічними процесами потребує швидкої обробки великого обсягу даних.
4. **Аналітика та оптимізація:** Здатність до аналізу даних, виявлення паттернів та можливостей для оптимізації технологічних процесів є важливою для підвищення продуктивності та якості продукції. Розробка алгоритмів аналізу та візуалізації даних відіграє ключову роль.
5. **Адаптація до змін:** В умовах швидкої зміни технологій та вимог ринку важливо мати гнучкі системи управління технологічними процесами, які можуть швидко адаптуватися до нових вимог та технологій.

Інтеграція технологій у веб-системах для управління технологічними процесами – це складний процес, оскільки вимагає співпраці різних компонентів та забезпечення їх взаємодії. Ось деякі ключові аспекти цієї проблеми:

1. **Стандартизація та сумісність:** Різні технології можуть використовувати різні протоколи, формати даних та архітектурні підходи. Для успішної інтеграції необхідно визначити стандартизовані протоколи обміну даними та забезпечити сумісність між системами.
2. **API та відкритість:** Використання відкритих API (інтерфейсів програмування додатків) дозволяє різним системам взаємодіяти між собою. Проте, інтеграція може бути складною, якщо API не дотримується стандартів або не є достатньо документованим.
3. **Міграція даних та синхронізація:** При інтеграції потрібно вирішити питання міграції даних з однієї системи до іншої та забезпечити їх постійну синхронізацію, щоб уникнути втрати даних або конфліктів.
4. **Безпека і доступність:** Під час інтеграції систем, важливо забезпечити безпеку передачі та зберігання даних, а також надійну доступність сервісів для уникнення витоку даних чи перерв у роботі систем.
5. **Тестування та моніторинг:** Після інтеграції необхідно провести тестування для перевірки правильності взаємодії між системами. Також важливо встановити системи моніторингу для виявлення проблем та реагування на них вчасно.

Стандартизація та сумісність - це ключові аспекти при інтеграції технологій, таких як React (фреймворк для розробки веб-інтерфейсів) та SheetDB (база даних). React може взаємодіяти з базою даних SheetDB через REST API. Цей підхід відповідає стандартам веб-розробки та дозволяють забезпечити сумісність між фронтендом (React) та базою даних (SheetDB).

Для взаємодії з базою даних SheetDB у React можна використовувати офіційні інструменти, наприклад, спеціальні драйвери або бібліотеки, які забезпечують стандартизований інтерфейс. Це дозволяє здійснювати ефективну взаємодію з базою даних за допомогою React.

Важливо правильно спроектувати модель даних для системи, щоб забезпечити відповідність між тим, як дані зберігаються у базі та їх використанням у React. Використання офіційної документації та інструментів для SheetDB допоможе забезпечити правильну інтеграцію та використання методів, що відповідають стандартам.

Крім цього, необхідно враховувати питання контролю версій та оновлень для забезпечення сумісності між версіями бібліотек та інструментів під час розширення функціоналу системи. Правильне управління версіями та оновленнями допоможе уникнути можливих конфліктів та забезпечить плавну роботу системи.

API (інтерфейс програмування додатків) - це набір правил та інструкцій, які дозволяють різним програмам чи сервісам взаємодіяти між собою. У контексті розробки веб-систем API використовується для обміну даними між фронтендом та сервером чи базою даних. Він визначає, як саме можна звертатися до системи, які дані можна отримати чи відправити, та у якому форматі це робити.

Важливою є також відкритість API. Це означає, що вони доступні для використання іншими розробниками без обмежень, дозволяючи створювати додатки чи розширення, які можуть взаємодіяти з вашою системою. Чим більша відкритість API, тим більше можливостей для інтеграції з іншими сервісами чи розробкою розширень.

У випадку React та SheetDB, використання відкритих API дозволяє реалізувати ефективну взаємодію між фронтендом та базою даних, дозволяючи отримувати необхідну інформацію та відправляти дані для подальшої обробки чи зберігання. Це відкриває широкі можливості для розширення функціоналу системи та інтеграції з іншими додатками чи сервісами.

Міграція даних та синхронізація в контексті веб-систем означає перенесення даних з одного джерела або формату до іншого так, щоб забезпечити їхню цілісність та доступність для подальшого використання в системі. Це важливий процес при зміні формату даних, оновленні бази даних чи переході до нових інструментів чи сервісів.

Під час міграції даних необхідно враховувати кілька ключових аспектів:

1. **Цілісність даних:** Перенесені дані повинні бути коректними та цілісними після міграції. Це означає перевірку та впевненість у тому, що дані не втрачені та не пошкоджені під час процесу перенесення.
2. **Збереження формату та структури:** Після міграції дані повинні зберегти свій формат та структуру для подальшого використання в системі. Це важливо, щоб уникнути проблем при подальшій обробці даних.
3. **Час та витрати:** Міграція може займати час та вимагати відповідних витрат ресурсів, особливо якщо йдеться про великі обсяги даних. Планування та оцінка ризиків допомагають уникнути можливих затримок чи проблем.
4. **Синхронізація:** Після міграції важливо забезпечити синхронізацію даних між відповідними системами чи компонентами, що використовують ці дані. Це дозволяє уникнути розбіжностей чи конфліктів у використанні даних.

Міграція даних та синхронізація - це складний, але необхідний процес для забезпечення правильного та безперебійного функціонування системи при зміні умов чи інструментів.

Безпека та доступність є критичними аспектами у веб-системах, особливо коли йдеться про управління технологічними процесами.

Безпека включає заходи, спрямовані на захист інформації від несанкціонованого доступу, зміни, чи втрати. У веб-орієнтованих системах це означає застосування шифрування для захисту даних, впровадження механізмів аутентифікації та авторизації, а також регулярне оновлення систем та програмного забезпечення для усунення вразливостей.

Доступність стосується можливості системи бути доступною користувачам у будь-який час, безперервно та без збоїв. Це означає розробку системи з урахуванням резервування та відновлення під час можливих збоїв, розподілу навантаження для уникнення перевантаження системи та забезпечення постійної доступності для користувачів.

У контексті управління технологічними процесами, безпека є важливою для захисту конфіденційної інформації про виробництво та процеси від несанкціонованого доступу. Доступність, з свого боку, забезпечує стабільну та безперебійну роботу системи у критичних ситуаціях, що може вплинути на виробничий процес. Обидва аспекти вимагають постійного моніторингу та підтримки для забезпечення найвищого рівня ефективності та безпеки системи.

Тестування та моніторинг є критичними складовими для забезпечення працездатності, ефективності та безпеки веб-системи для управління технологічними процесами.

Тестування включає проведення різноманітних тестів, які оцінюють працездатність системи та її компонентів. Це може бути функціональне тестування, яке перевіряє, чи виконуються очікувані функції системи, або навантажувальне тестування, яке визначає, як система працює при великому навантаженні. Тестування безпеки та вразливостей також важливе для виявлення та виправлення можливих проблем з безпекою.

Моніторинг включає постійне спостереження за роботою системи, її продуктивністю та безпекою. Це означає встановлення метрик та індикаторів продуктивності, які дозволяють відслідковувати роботу системи в реальному часі. Моніторинг допомагає вчасно виявляти проблеми та збої, реагувати на них та запобігати можливим аваріям.

У випадку управління технологічними процесами, тестування дозволяє переконатися, що система правильно взаємодіє з технологічним обладнанням та обробляє дані належним чином. Моніторинг у свою чергу дозволяє вчасно виявляти будь-які неполадки у процесах та забезпечує постійну доступність системи для оптимального управління технологічним процесом. Обидва процеси, тестування та моніторинг, є важливими для забезпечення ефективності та стабільності веб-системи.

2.1. Аналіз існуючих підходів до керування технологічним процесом

Аналіз існуючих підходів до керування технологічним процесом - це важлива частина при розробці системи для управління технологічними процесами. Це передбачає дослідження та оцінку різних методів, стратегій та підходів, які використовуються у сучасних технологічних системах.

Такий аналіз може включати вивчення та порівняння різних методологій керування процесами, таких як Lean Manufacturing, Six Sigma, а

також використання сучасних технологій у цілях автоматизації та оптимізації виробничих процесів.

Під час аналізу важливо враховувати специфіку технологічних процесів у конкретній галузі чи виробництві, щоб знайти найбільш ефективні та підходящі методи та практики.

Такий аналіз надасть можливість вибрати оптимальний підхід до управління технологічними процесами для подальшої імплементації у систему, що розробляється, забезпечуючи більш ефективне та оптимізоване управління виробничими процесами на підприємстві чи у відповідній галузі.

Lean Manufacturing та Six Sigma є двома визнаними методологіями управління якістю та оптимізації процесів у виробництві, а використання сучасних технологій автоматизації додає нові можливості у цьому контексті.

Lean Manufacturing базується на принципах оптимізації виробничих процесів шляхом усунення зайвого та мінімізації втрат. Вона спрямована на забезпечення максимальної ефективності за рахунок виявлення та усунення марнощів у виробничому процесі, таких як очікування, надмірний запас, зайві рухи тощо.

Six Sigma фокусується на покращенні якості продукції та оптимізації процесів через систематичний аналіз та виправлення дефектів, а також мінімізацію відхилень від стандартів якості.

Обидві методології спрямовані на покращення ефективності та якості виробництва, але Lean зосереджується на усуненні марнощів, тоді як Six Sigma більше націлена на статистичний аналіз процесів.

Використання **сучасних технологій** у виробництві дозволяє автоматизувати багато процесів та використовувати інноваційні рішення для

оптимізації виробничих процесів. Це може включати в себе використання Інтернету Речей (IoT) для моніторингу, штучний інтелект для прогнозування та управління, розумні системи управління запасами, автоматизовані лінії виробництва тощо.

Злагоджений підхід, який поєднує Lean та Six Sigma з сучасними технологіями автоматизації, може дати комплексний ефект у плані оптимізації, підвищення якості та ефективності виробництва.

Lean Manufacturing і Six Sigma - це, в першу чергу, стратегії та методи управління, спрямовані на оптимізацію та покращення виробничих процесів.

Lean Manufacturing фокусується на ефективному управлінні виробничими процесами, усуненні зайвого та мінімізації втрат, що може бути сприяне через впровадження різноманітних методів, таких як "Kaizen" (постійне покращення) або "Just-in-Time" (точно вчасно), серед інших.

Six Sigma, з іншого боку, акцентує на статистичному аналізі процесів для мінімізації дефектів та відхилень від стандартів якості. Це дозволяє виявляти, аналізувати та виправляти недоліки у виробничих процесах, що допомагає підвищувати якість продукції та ефективність виробництва.

Обидва підходи можуть бути успішно впроваджені як у виробничій, так і у послуговій сферах, та сприяють забезпеченню оптимального управління та покращенню процесів у компаніях.

Аналіз існуючих підходів до керування технологічними процесами є ключовим етапом для вибору оптимальних стратегій управління виробничим процесом. У контексті React та SheetDB, варто звернути увагу на специфічні можливості цих інструментів та їхній вплив на управління технологічними процесами. Комбінація React та SheetDB може створювати потужну

інфраструктуру для візуалізації даних, моніторингу та управління технологічними процесами. Під час аналізу підходів до управління цими процесами, важливо врахувати можливості та обмеження кожного інструменту та їх потенційні переваги у створенні ефективних систем управління.

- **React** — це бібліотека JavaScript для побудови інтерфейсів користувача, яка дозволяє створювати динамічні та ефективні веб-додатки. У контексті управління технологічними процесами, React може використовуватися для створення веб-інтерфейсу для моніторингу та управління цими процесами. Він дозволяє реалізувати інтерактивність та швидкість роботи, що є важливими аспектами для ефективного керування технологічними процесами.
- **SheetDB** — це онлайн-сервіс, що надає можливість зберігати дані в онлайн-таблицях, що функціонують як база даних. У контексті управління технологічними процесами, SheetDB може використовуватися для зберігання та організації даних, які відносяться до цих процесів. Завдяки його можливостям можна легко звертатися до даних та виконувати операції з ними через веб-сервіс.

Переваги використання Excel для зберігання формул та даних включають:

1. **Зручність інтерфейсу:** Excel має інтуїтивний інтерфейс, який дозволяє швидко створювати формули та працювати з даними без потреби в складних програмах.
2. **Великий функціонал:** В Excel доступний широкий спектр функцій та формул для обробки даних, що дозволяє проводити різноманітний аналіз і операції з даними.
3. **Швидкість та локальне збереження:** Він працює на рівні локальної машини, тому операції можуть виконуватись швидше порівняно з

онлайн-базами даних. Крім того, можливість роботи в автономному режимі.

Недоліки включають:

1. **Обмежена масштабованість:** При рості обсягу даних чи складності формул, Excel може стати обмеженням у швидкості роботи та обробці даних.
2. **Брак системної безпеки:** Дані в Excel можуть бути вразливі до втрати, особливо якщо вони не захищені від випадкового видалення або збоїв в програмі.
3. **Брак колективної роботи:** При спільній роботі над документом в Excel можуть виникати проблеми з одночасним редагуванням, що може призвести до конфліктів даних.

Однак важливо враховувати, що використання Excel як засобу збереження формул та даних залежить від конкретних потреб та розмірів проекту. Для великих обсягів даних або потреби в більшій безпеці та колективній роботі, можливо, буде кращим вибором використання спеціалізованих баз даних, таких як SheetDB або MongoDB.

Excel має безліч переваг, зокрема зручний інтерфейс та розгалужені можливості обробки даних і створення формул. Його широкий функціонал дозволяє виконувати складні операції та аналіз, що робить його корисним для багатьох завдань. Однак у великих обсягах даних чи у випадку потреби у спільній роботі з кількома користувачами Excel може виявитися неефективним.

Наприклад, коли йдеться про колективне редагування, можливі конфлікти, які важко вирішити. Також, Excel не є найкращим вибором для

зберігання великих обсягів даних, оскільки його продуктивність може впливати на швидкодію при обробці великих таблиць.

У випадку, якщо потрібно зберігати дані у веб-середовищі та мати можливість легко спільно працювати з даними, онлайн-бази даних, такі як SheetDB або MongoDB, можуть бути більш ефективними. Ці інструменти можуть забезпечити кращу безпеку даних, масштабованість та можливості для колективної роботи, що робить їх вигідними для використання у великих проектах чи командних зусиллях.

Розгляньмо відмінності між MongoDB та SheetDB у контексті роботи з великими формулами та обчисленнями. MongoDB - це документоорієнтована база даних, яка зазвичай використовується для зберігання даних у форматі JSON-подібних документів. У MongoDB, збереження складних обчислень або великих формул може потребувати перетворення цих обчислень у структури JSON, що може бути важким та неефективним.

У разі, коли у вас є великі обсяги даних або складні обчислення, SheetDB може стати більш зручним рішенням. SheetDB працює як онлайн-таблиця з можливістю зберігання та організації даних у форматі, схожому на електронну таблицю. Це означає, що ви можете працювати з даними, що виглядають як звичайна таблиця, без необхідності перетворення їх у специфічні формати, що вимагається у MongoDB. Великим плюсом є можливість простої інтеграції, зручність у використанні та можливість прямої роботи з даними у вигляді таблиць.

SheetDB також може бути зручним в контексті складності обчислень. У випадках, коли формули або обчислення складні або великі, вони можуть бути важкими для управління та зберігання у вигляді JSON-структур у MongoDB, але вони можуть бути легко розміщені та організовані в таблиці у SheetDB.

Отже, хоча MongoDB є потужною базою даних, для великих обсягів даних та складних обчислень SheetDB може бути більш придатним інструментом завдяки своїй простоті у використанні та можливості працювати з даними у формі таблиць без складних перетворень. SheetDB і MongoDB мають різні підходи до зберігання даних. MongoDB - це база даних NoSQL, яка зберігає дані у форматі JSON-подібних документів, що дозволяє гнучко структурувати інформацію. Однак, коли мова йде про складні обчислення або великі обсяги даних з великою кількістю формул, прив'язка цих даних до структури бази даних MongoDB може бути складною і вимагати додаткової обробки.

З іншого боку, SheetDB пропонує можливість зберігати дані у формі електронної таблиці онлайн. Це може бути зручним в разі великої кількості даних або великої кількості формул. Ви можете працювати з цими даними безпосередньо у вигляді таблиць, що значно спрощує їхнє управління та обробку.

У разі використання складних формул та обчислень, SheetDB може бути зручнішим вибором через здатність зберігати та організовувати дані у зрозумілому форматі. Це дозволяє швидше та легше працювати з даними та виконувати обчислення без необхідності конвертації даних у специфічний формат, як це потрібно у випадку MongoDB.

Таким чином, вибір між цими базами даних залежить від конкретних потреб: MongoDB може бути потужним для складних структур даних, а SheetDB - для зручності роботи з великими обсягами даних та складними формулами.

2.2. Огляд баз даних, зокрема SheetDB, для збереження та обробки формул

SheetDB - це сервіс, який надає можливість зберігання даних у форматі таблиць, що нагадує електронну таблицю, проте операційний в середовищі онлайн. Цей сервіс має кілька переваг для зберігання та обробки формул:

1. **Простота використання:** Його інтерфейс подібний до електронних таблиць, що робить його дуже зручним для користувачів, оскільки вони знайомі з роботою у такому середовищі.
2. **Легкість збереження та організації даних:** SheetDB дозволяє організувати дані у формі таблиць, що спрощує їхнє збереження та структурування. Ви можете створювати різні таблиці для різних типів даних, що полегшує управління проектами.
3. **Миттєвий доступ та онлайн-робота:** Оскільки SheetDB працює онлайн, ви можете отримати доступ до даних у будь-який час та з будь-якого пристрою, який має підключення до Інтернету.
4. **Підтримка формул:** SheetDB підтримує використання формул для обчислення значень в різних комірках таблиць, що дозволяє проводити розрахунки безпосередньо в таблицях, спрощуючи аналіз та обробку даних.
5. **Робота з API:** Вона має можливість використання API для збереження та витягування даних, що відкриває можливості для автоматизації процесів і інтеграції з іншими сервісами чи програмами.
6. **Широкі можливості колаборації:** SheetDB дозволяє спільно працювати над даними з іншими користувачами, що є важливим для командної роботи над проектами.

7. **Сумісність з багатьма програмами:** SheetDB може бути інтегрована з багатьма програмами та інструментами через API. Це відкриває можливості для автоматизації та обміну даними між різними платформами.
8. **Безпека даних:** SheetDB забезпечує захист даних через шифрування та інші заходи безпеки, забезпечуючи конфіденційність та цілісність інформації.
9. **Масштабованість:** При необхідності SheetDB може масштабуватися, щоб відповідати зростаючим потребам у зберіганні даних, дозволяючи розширювати кількість даних та користувачів, якщо це необхідно для вашого проекту.
10. **Підтримка користувача:** SheetDB може надавати підтримку користувачам у разі виникнення питань чи проблем з використанням сервісу.
11. **Зручний інтерфейс:** Інтерфейс SheetDB призначений для зручного використання, що дозволяє швидко створювати таблиці, додавати та організовувати дані.
12. **Автоматизація рутинних завдань:** SheetDB дозволяє автоматизувати рутинні завдання за допомогою скриптів та запитів до API. Це спрощує процес роботи з даними та може прискорити робочий процес.
13. **Гнучкість у роботі з даними:** Ви можете використовувати SheetDB для збереження різних типів даних, від числових значень до текстової інформації, що робить його універсальним інструментом для різних потреб.
14. **Спільна робота:** Інша перевага - це можливість спільно працювати з даними, ділитися доступом до таблиць та змінювати їх разом з іншими користувачами, що спрощує колаборацію в команді.

15. Економія часу та ресурсів: SheetDB може допомогти економити час та ресурси, оскільки він забезпечує швидкий доступ до даних, спрощує процес їхньої організації та аналізу.

Загалом, SheetDB є зручним інструментом для збереження та обробки даних у вигляді таблиць, надаючи широкий функціонал, масштабованість, спільну роботу та можливість автоматизації процесів.

2.2.1. Кроки для початку користування SheetDB:

Авторизація та робота з акаунтом:

Після того як ми зайшли на SheetDB зареєстрували і авторизувалися нам потрібно створити API та підключити нашу Excel таблицю до SheetDB, для того щоб це зробити потрібно:

На головній сторінці натиснути кнопку Create new API після чого ми попадаємо на сторінку (див. рис. 1.1)

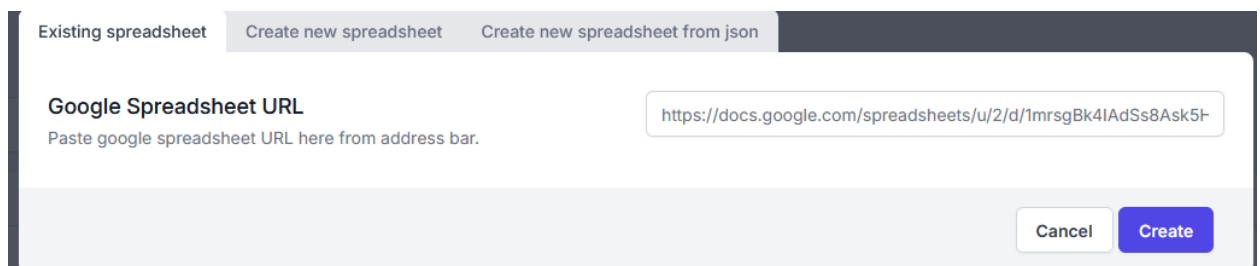


Рисунок 2.1 – Авторизація та робота з акаунтом

Після чого нам потрібно вставити в відповідне поле нашу ссилку на нашу таблицю і після чого ми зможемо працювати з нею

2.3. Структура та архітектура веб-системи для керування технологічним процесом

Структура та архітектура веб-системи для керування технологічним процесом може бути складною, оскільки вона повинна обслуговувати різноманітні функції та процеси. Ось кілька ключових аспектів, які слід врахувати при проектуванні такої системи.

2.3.1. Компоненти системи:

1. **Frontend (Клієнтська частина):** Веб-інтерфейс, який взаємодіє з користувачем. Він може містити панелі управління, форми для введення даних, графіки та звіти.
2. **Backend (Серверна частина):** Це сервер, який обробляє запити від клієнтської частини. Він відповідає за обробку даних, бізнес-логіку, взаємодію з базою даних та іншими сервісами.

Архітектурні підходи:

1. **MVC (Model-View-Controller):** Розділення системи на моделі (дані), представлення (інтерфейс) та контролери (логіка) для забезпечення гнучкості та організації коду.
2. **Microservices або Сервіс-орієнтована архітектура:** Розбиття системи на невеликі незалежні компоненти, які можуть працювати окремо, спрощуючи розгортання та масштабування.

Безпека та Автентифікація:

1. **Автентифікація та Авторизація:** Забезпечення безпеки даних через правильну аутентифікацію користувачів та управління доступом до функцій системи.

2. **Шифрування даних:** Застосування шифрування для захисту конфіденційної інформації в базі даних та під час передачі даних між клієнтом та сервером.

База даних:

1. **Вибір БД:** Потрібно обрати базу даних, яка відповідає потребам системи. Це може бути MongoDB, як у вас раніше згадувалося, або інша БД залежно від потреб.
2. **Оптимізація даних:** Створення оптимізованої моделі даних для ефективного збереження та маніпуляції інформацією.

Вибір технологій для розробки веб-сайту, зокрема фреймворків, мов програмування тощо

Фронтенд:

1. **Користувацький Інтерфейс (UI):** Розробка інтерфейсу, що відображає дані користувачам. Може включати форми введення, графіки, таблиці та інші елементи.
2. **Функціональність:** Реалізація логіки, яка відповідає за взаємодію з даними та виконання певних операцій на стороні клієнта.
3. **Реактивність та Відгук Користувача:** Забезпечення швидкої та реактивної відповіді на дії користувача, щоб забезпечити плавну роботу веб-системи.

База даних (SheetDB):

1. **Збереження та Організація Даних:** SheetDB дозволяє створювати та управляти таблицями, в яких можна зберігати дані.
2. **Робота з Даними через API:** Доступ до даних через API, що дозволяє зчитувати, записувати та оновлювати дані з клієнтської сторони.

3. **Гнучкість та Швидкість:** Зручність роботи з даними та швидкість доступу до них для ефективної роботи фронтенду.

Інтеграція Фронтенду та Бази Даних:

1. **Запити на Дані:** Фронтенд взаємодіє з базою даних через API SheetDB, відправляючи запити для отримання, оновлення або збереження даних.
2. **Управління Даними:** Функції фронтенду виконують операції з даними, наприклад, введення нових даних користувачем або відображення інформації, що приходить з бази даних.
3. **Оптимізація Даних та Запитів:** Розробка фронтенду та запитів до бази даних таким чином, щоб максимально оптимізувати швидкість та ефективність обміну даними.

React став моїм вибором для фронтенду через його універсальність, широкий вибір бібліотек та зручний JSX синтаксис, що дозволяє створювати компоненти швидко та ефективно, його використання дозволило мені зосередитися на створенні високоякісного користувацького інтерфейсу з великою кількістю відмінних компонентів та можливостей для створення динамічних інтерактивних елементів. Багатофункціональний характер React сприяв зручній та швидкій розробці, особливо при роботі з обсягом даних та важкою логікою. Його модульна структура дозволяє розбивати інтерфейс на невеликі частини, що спрощує розробку та робить код більш організованим.

Зараз зосереджений на використанні SheetDB для зберігання даних. Вибір був зроблений через його простоту інтерфейсу та зручність роботи з API. Його простота та легкість використання були великим плюсом. Можливість працювати з даними у вигляді таблиць через API забезпечила мені зручність у взаємодії з даними безпосередньо з фронтенду. Такий підхід спрощує використання та обробку даних, дозволяючи швидко та ефективно взаємодіяти з інформацією, необхідною для проектування інтерфейсу та виконання різних операцій на клієнтській стороні.

Це дозволяє безпосередньо взаємодіяти з даними у вигляді таблиць, забезпечуючи гнучкість та легкість управління інформацією. Такий підхід дозволяє легко і швидко отримувати доступ до даних з фронтенду та ефективно використовувати їх для відображення та виконання операцій. Крім того, можливість масштабування SheetDB відкриває можливості для роботи з різними обсягами даних та різними масштабами проектів.

РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ ТА ТЕСТУВАННЯ

3.1. Розробка інтеграції Excel формул до бази даних SheetDB

Отже після того як я проаналізував і зрозумів що мені потрібно для того щоб реалізувати систему-керування технологічним процесом давайте розглянемо що конкретно і як потрібно це зробити. Після того як ми зареєструвалися і зайшли авторизувалися на SheetDB і створили нашу API із нашою таблицею нам потрібно ознайомитися з документацією як і що нам потрібно. SheetDB у своїй документації пропонує нам такий вибір (див. рис. 3.1).

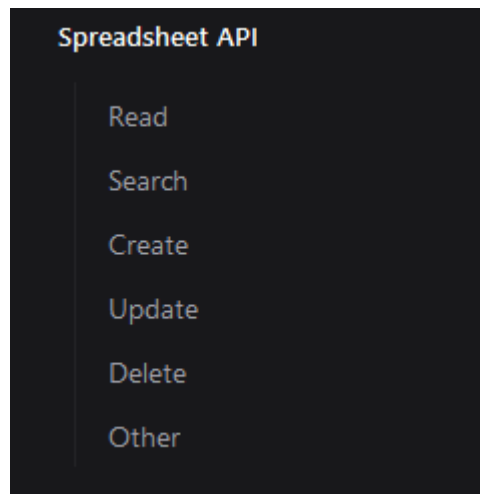


Рисунок 3.1 – Документація API

Отже API нам пропонує такі функції як прочитання нашої таблиці пошук, створення нової таблиця, оновлення таблиці та видалення з таблиці в нашому випадку нам потрібно усього лиш дві функції це **Update** – тобто оновлення клітинок у таблиці, та **Read** – для того щоб можна було виводити результати користувачу після того як він відправить запит. Розберемо кожен з методів ближче візьмемо функцію **Update** (див. рис 3.2)

Update

PATCH · /api/v1/{API_ID}/{COLUMN_NAME}/{VALUE}

Update with single query

Updates the content for the specified row. You must specify a column name and a value to find the row(s) you want to update. All rows that meet the condition will be updated. It updates only the columns passed in the `data` object, the other columns will remain unchanged.

Returns count of updated rows.

Required attributes

data json

The object containing the data you want to change.

Example: `{'name': "Mark", 'age': 28}`

Optional attributes

sheet string

The sheet (tab) you want to select.

mode string

Value Input Option, [more info here](#).

Request

cURL JavaScript PHP

PATCH · /api/v1/{API_ID}/{COLUMN_NAME}/{VALUE}

```
fetch('https://sheetdb.io/api/v1/58f61be4dda40/id/3', {
  method: 'PATCH',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    data: {
      'name': "Emma"
    }
  })
})
.then((response) => response.json())
.then((data) => console.log(data));
```

JSON Response

```
{
  updated: 1
}
```

Рисунок 3.2 – метод Update

Для того щоб імплементувати цю функцію у наш сайт нам потрібно зрозуміти що власне нам потрібно для того щоб оновлювати наші комірки у Excel а для цього нам для початку потрібен наш API-id який знаходиться у нашому API який ми створили на SheetDB (див. рис. 3.3)

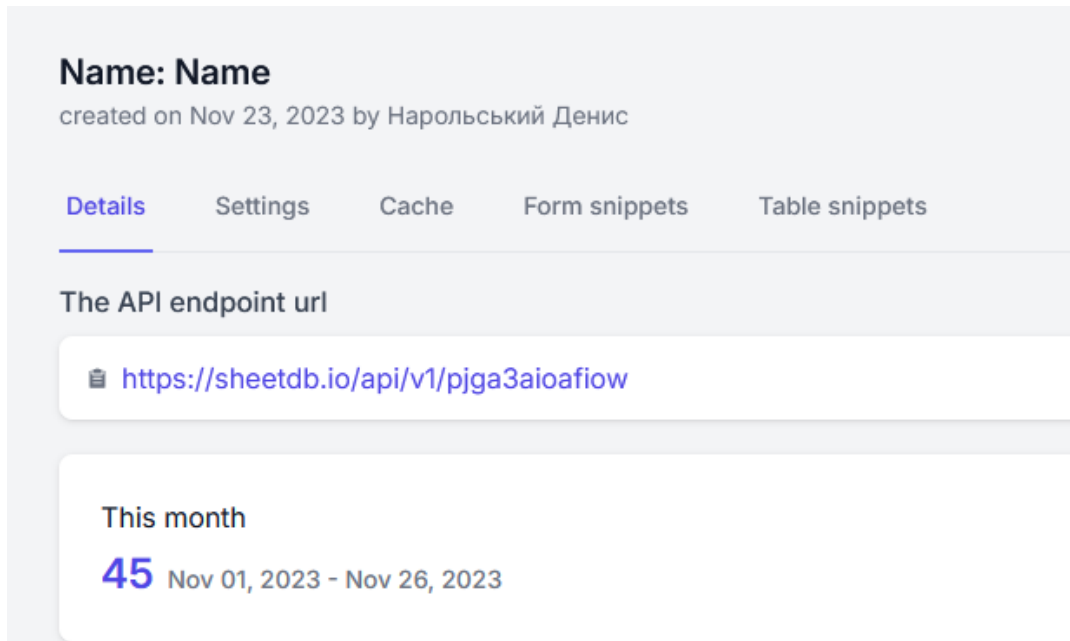


Рисунок 3.3 – Ендпоінт API

Після чого нам потрібне ім'я нашої колонки і те значення яке ми хочемо замінити але десятки спроб оновити значення ось таким шляхом не принесли ніякого результату тому що початкове значення в нас йде 0, і тому такий спосіб для мене не підходить тому що воно оновлює тільки унікальне value.

У моєму випадку я вирішив зробити це більше простим способом як на мене і більш ефективним, що я для цього зробив створив у своєму Excel файлу новий аркаш з назвою backend (див. рис. 3.4.)

№ з/п	Найменування робіт	Одиниці виміру	Ціна за одиницею (грн.)	94444	Робочого часу (год.)	Всього (грн.)
0	Набір тексту (формат А4)	маш. ст.	0,0000	777	0,00	0,00
1	Підготовка до виходу в світ	стор. А5	0,0000	929	0,00	0,00
2	Макетування книги	стор. А5	0,0000	929	0,00	0,00
3	Макет обкладинки	стор. А5	0,0000	929	0,00	0,00
4	Векторна графіка	стор. А4	0,0000	070	0,00	0,00

Рисунок 3.4 – Новий аркуш для отримання інформації

І ось тепер після того як ми створили новий аркуш в нас є дуже простий спосіб як можна обробляти нашу інформацію з сайту і вносити її до комірок ми просто будемо створювати нові значення і зв'язувати їх з таблицею з формулами для того щоб відправити нові значення в новий аркуш нам потрібна функція **Create**(див. рис. 3.5)

Create

`POST` · `/api/v1/{API_ID}`

Create

Creates row(s) in the spreadsheet. Your request should contain a `data` attribute - it should be an array of rows. The keys inside the object should be the column names, and the values will be filled in the spreadsheet. The rows will be added at the end of the sheet.

You can use `INCREMENT` value. SheetDB will look for the biggest number in the given column and increase by 1.

You can use the `TIMESTAMP` value to use the current timestamp (in unix time).

You can use the `DATETIME` value to use the current date
Format: `2016-06-23 09:07:21`.

If for some reason you can't put your data in the `data` attribute, SheetDB will try to use the entire request data to match the spreadsheet columns.

API will return the number of created rows with status code `201 Created`.

Required attributes

`data` `json`

An array with the rows you want to append.
Example: `[{'id': 4, 'name': "Mark"}, {'id': 5, 'name': "Susan"}]`

Request

cURL [JavaScript](#) [PHP](#)

`POST` · `/api/v1/{API_ID}`

```
fetch('https://sheetdb.io/api/v1/58f61be4dda40', {
  method: 'POST',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    data: [
      {
        'id': "INCREMENT",
        'name': "Mark",
        'age': 18
      }
    ]
  })
})
.then((response) => response.json())
.then((data) => console.log(data));
```

JSON Response

```
{
  created: 1
}
```

Рисунок 3.5 – Функція Create

Тут приблизно така сама логіка як і в функції **Update** тільки ми просто задаємо нашу API-Id і задаємо додатковий параметр в запиті такий як **sheet** він опціональний і відповідальний за те на який аркуш буде здійснюватися той чи інший запит.

3.2. Програмування та реалізація веб-інтерфейсу для роботи з цими даними

Тепер ми маємо усе необхідне для того щоб створити веб-інтерфейс та логіку з-за допомогою якої ми будемо здійснювати запити на нашу базу даних і вираховувати формули, для початку створимо функцію з-за допомогою якої ми будемо добавляти в нашу базу даних значення. (рис. 3.6)

```

fetch("https://sheetdb.io/api/v1/pjga3aioafiw", {
  method: "POST",
  headers: {
    Accept: "application/json",
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    sheet: "backend", // вказуємо назву аркуша
    data: [
      {
        id: "INCREMENT",
        name: "John",
        age: 25,
      },
    ],
  }),
})
  .then((response) => response.json())
  .then((data) => console.log(data));

```

Рисунок 3.6 – Функція для додавання в базу даних значень

Для початку спробуємо відправити статичні данні та подивимось чи вони будуть зберігатися у нашій базі даних для цього цю функцію вішаємо на кнопку і натискаємо її.

У консолі ми бачимо що об'єкт був створений а отже дані успішно внеслись в нашу базу даних (див. рис. 3.7)

```

▶ Object { created: 1 }
>>

```


Рисунок 3.7 – Зображення консолі

Переходимо на нашу базу даних та звіряємо чи відправились данні до неї(див. рис. 3.8).

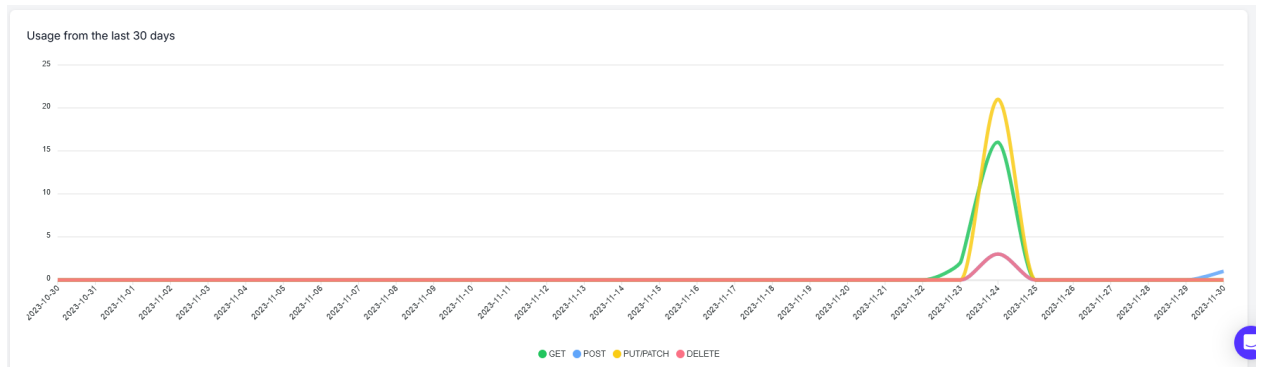


Рисунок 3.8 – Графік запитів до бази даних

Потім переходимо на аркуш backend і дивимось чи прийшли наші дані (див. рис. 3.9)

id	name	age
1	John	25
2	John	25
3	John	25

Рисунок 3.9 – Аркуш backend

Як ми бачимо з цього скріншоту дані прийшли та збереглись на нашому аркуші, після чого з-за допомогою функцій Excel зв'язуємо рядки які нам потрібні.

Після того як ми протестували запит потрібно створити користувацький інтерфейс з-за допомогою якого користувачі будуть відправляти на нашу серверну частину данні для подальшої їх обробки та виведення результатів. Для цього ми прив'язуємо з-за допомогою кода логіку для обробки цих даних до наших полів і кнопок. Відправляємо тестовий запит і перевіряємо чи все працює. Дивимось що порахувало нам наша база даних та бачимо що вартість становить 11 001,58 грн (див. рис. 3.10)

Вартість матеріалів						
	Матеріал	Густина /%	К-ість пакек	Кількість (кг)	Ціна за кг	Всього (грн.)
32	Папір А3 (500 арк./паку)	65	0,00	0,00	0,00	0,00
33	Папір А4 (500 арк./паку)	80	0,00	0,00	0,00	0,00
34	Картон на обкладинку А3 (250 арк./паку)	160	0,00	0,00	0,00	0,00
35	Картон на обкладинку А4 (250 арк./паку)	160	0,00	0,00	0,00	0,00
Сума:						0,00

<p>Наклад: 100</p> <p>Наклад без контрольної розсилки: 85</p> <p>Коштом Замовника: 85</p> <p>Коштом Виконавця: 0</p> <p>Контрольна розсилка: 15</p> <p>На папері Замовника: 0</p> <p>На поліграфічних матеріалах Замовника: 0</p>	<p>Всього: 11 001,58 грн.</p> <p>Собівартість: 129,43 грн.</p> <p>Вартість для Замовника: 11 001,58 грн.</p> <p>Вартість для Виконавця: 0,00 грн.</p>
---	---

Калькуляцію витрат перевірів
заступник головного бухгалтера
Т. В. Качинська

Рисунок 3.10 – Результати обчислення формул

Після чого ми відправляємо запит до клієнта і в нього появляється можливість натиснути на кнопку та йому видасть вартість його замовлення (див.рис.3.11)

Шановні пані та панове!

Ми раді, що Ви звернулись до нас, і зробимо все можливе, щоб співробітництво з компанією «Polygraphu» було приємним і взаємовигідним. Якщо Вас зацікавила продукція нашого підприємства, **заповніть форму замовлення.**

ФОРМА ЗАМОВЛЕННЯ

Ім'я	Прізвище
По батькові	Посада
Поштова адреса	Номер телефону
Вид видання	Назва видання
Кількість сторінок	Кількість примірників
Додаткова інформація	

Вартість замовлення буде становити: 11 001,58 грн

[РЕЗУЛЬТАТ](#)

Рисунок 3.11 – Вивід результату клієнту

ВИСНОВКИ

Розвиток інтернет-технологій неодмінно перетворює спосіб, яким ми сприймаємо та використовуємо інформацію. Веб-сайти стають не лише візитними картками сучасних підприємств, а й потужними інструментами відкриття нових можливостей та розширення географії їх діяльності.

Розроблена в рамках дипломного проекту web-орієнтована система керування технологічним процесом для поліграфічного підприємства є відповіддю на потреби сучасного ринку. Її впровадження відкриває нові горизонти у покращенні ефективності виробництва та забезпеченні швидкого доступу до необхідної інформації.

Важливо зазначити, що розроблений веб-сайт стає не просто платформою для представлення продукції, але й інструментом, який дозволяє клієнтам отримати доступ до повного спектру послуг та товарів, навіть в онлайн-режимі. Його функціональні можливості забезпечують зручність та доступність для користувачів, що сприяє підвищенню обсягів замовлень та розширенню аудиторії підприємства.

Процес розробки веб-сайту включав в себе не лише застосування передових веб-технологій, але й глибокий аналіз потреб клієнтів та вимог сучасного ринку. Результатом став продукт, який в повній мірі відповідає поставленим завданням та готовий до активного використання.

У цілому, розробка цієї web-орієнтованої системи для поліграфічного підприємства відображає важливість використання сучасних технологій у сфері бізнесу та демонструє потенціал їхнього впливу на підвищення ефективності та конкурентоспроможності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Stavrovskiy, A.B., Karnaukh, T.O. "Pershi kroky prohramuvannia." K.: Dialektika, 2005. 389 p.
2. Minnick, Chris, Holland, Eva. "JavaScript for Beginners" 2019. 320 p.
3. Brown, Ethan. "Learning JavaScript: A Guide to Building Modern Websites." 2016. 368 p.
4. Electronic resource [Accessed]: URL: <https://reactjs.org/>
5. Electronic resource [Accessed]: URL: <https://javascript.info/>
6. Electronic resource [Accessed]: URL: <https://uk.wikipedia.org/wiki/MongoDB>
7. Robbins, Jennifer. "HTML5 Pocket Reference." O'Reilly Media; Fifth edition (September 10, 2013). 182 p.
8. Duckett, Jon. "HTML and CSS: Design and Build Websites." John Wiley & Sons; 1st edition (November 8, 2011). 490 p.
9. CMSList. "Overview of CMS." Website about site management systems. URL: <http://www.cmslist.ru>
10. Electronic resource [Accessed]: URL: <https://www.w3schools.com/nodejs/>
11. Electronic resource [Accessed]: URL: https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm
12. Freeman, Adam. "Pro React 16." Apress; 5th edition (December 21, 2019). 696 p.
13. Duckett, Jon. "JavaScript and JQuery: Interactive Front-End Web Development." John Wiley & Sons; 1st edition (July 8, 2014). 640 p.
14. Flanagan, David. "JavaScript: The Definitive Guide." O'Reilly Media; 7th edition (June 30, 2022). 868 p.
15. Electronic resource [Accessed]: URL: <https://developer.mozilla.org/enUS/docs/Web/JavaScript>
16. Flanagan, David. "Node.js: Up and Running." O'Reilly Media; 2nd edition (August 21, 2015). 334 p.

17. Hughes-Croucher, Shelley. "Node: Up and Running: Scalable Server-Side Code with JavaScript." O'Reilly Media; 1st edition (November 9, 2012). 204 p.
18. Electronic resource [Accessed]: URL: <https://expressjs.com/>
19. Shifflet, Todd. "Learning Node: Moving to the Server-Side." O'Reilly Media; 2nd edition (October 15, 2016). 332 p.
20. Electronic resource [Accessed]: URL: <https://www.mongodb.com/>
21. Welling, Luke. "MongoDB in Action." Manning Publications; 2nd edition (October 28, 2018). 560 p.
22. Nayrolles, David. "Mastering MongoDB 4.x: Expert techniques to run high-volume and fault-tolerant database solutions using MongoDB 4.x." Packt Publishing; 2nd edition (June 29, 2021). 534 p.
23. Electronic resource [Accessed]: <https://www.npmjs.com/>
24. Young, Aaron. "Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB." O'Reilly Media; 1st edition (August 28, 2015). 190 p.
25. Electronic resource [Accessed]: URL: <https://www.typescriptlang.org/>