МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра комп'ютерних наук та програмної інженерії

РЕФАКТОРИНГ РЕСУРСУ ФОТОГАЛЕРЕЇ З CMS WORDPRESS НА JS + GOOGLE API.

Кваліфікаційна робота (проєкт)

на здобуття ступеня вищої освіти "бакалавр"

Виконав: студент 4 курсу 12-441 групи
Спеціальності: 121 Інженерія
програмного забезпечення
Освітньо-професійної програми «Інженерія
програмного забезпечення»

Гавриленко Владислав Олегович

Науковий керівник: Співаковський О.В.
доктор педагогічних наук, професор
Рецензент:Кльонов Д.М. full stack
developer, фріланс

Херсон – 2024

Зміст

Вступ

У сучасному світі технологій, ефективність та технологічність веб-ресурсів відіграють значну роль у забезпеченні конкурентоспроможності та задоволенні потреб користувачів. Головна вимога у веб-розробці, особливо для фотогалерей, полягає в тому, щоб поєднати зручність управління контентом, швидкість завантаження та гнучкість і персоналізовану взаємодію з користувачем. Рефакторинг ресурсу фотогалереї з CMS Wordpress на сучасний стек технологій JS + Google API відкриває нові можливості для вдосконалення веб-ресурсів, роблячи їх більш зручними до вимог сучасного цифрового світу.

Незважаючи на те, що популярні платформи для управління контентом, такі як WordPress, пропонують широкі можливості для швидкого розгортання та управління веб-сайтами, вони можуть стати обмеженням для розвитку веб-ресурсів. Використання звичайних шаблонів і плагінів може зробити інтеграцію певних функцій, необхідних для конкретної фотогалереї, складніше, і може вплинути на швидкість завантаження сторінок і загальну продуктивність сайту. У таких ситуаціях необхідний рефакторинг, щоб використовувати більш сучасні та ефективні технології.

Головна мета рефакторингу полягає у переході від загальноприйнятих рішень на базі Wordpress до індивідуалізованої платформи, що використовує JavaScript та Google API. Цей перехід забезпечить більшу гнучкість для задоволення різних потреб і ідей, а також покращить швидкість роботи та інтерактивність веб-сайту. Сучасні технології пропонують безліч можливостей для оптимізації роботи веб-сайту, SEO-позиціонування та загального користувацького досвіду.

Швидкий розвиток веб-технологій, а також зростаючі вимоги користувачів до якості та функціональності веб-сайтів визначають актуальність теми рефакторингу веб-ресурсів з використанням CMS Wordpress на JS + Google API. Сучасні веб-сайти повинні бути швидкими, безпечними та адаптованими до різних пристроїв і платформ.

Задовольнити ці вимоги можна за допомогою рефакторингу та використанню сучасних технологій. Це надає власникам веб-сайтів переваги в конкурентній боротьбі між веб-сайтами та відкриває нові можливості для розвитку проектів, збільшуючи їх функціональність.

Розділ 1. Теоретичні основи рефакторингу

1.1 Визначення та цілі рефакторингу

Рефакторинг — це процес детальної оптимізації внутрішньої структури програмного забезпечення без впливу на його функціональну поведінку. Цей процес забезпечує більш ефективне внесення майбутніх оновлень, підвищення стабільності програми та оптимізацію її продуктивності, а також поліпшує читабельність коду та спрощену архітектуру. Основні цілі рефакторингу включають:

- Підвищення якості коду: забезпечення зрозумілості та ефективності програмного коду шляхом спрощення його структури та зменшення складності.

- Забезпечення гнучкості: підготовка коду до легкої інтеграції нових функцій чи технологій полягає у створенні структури коду, яка сприяє швидкому додаванню, або зміні функціоналу без необхідності переробки великої частини програми.

- Оптимізація продуктивності: підвищення ефективності програми за допомогою оптимізації її внутрішньої логіки та структури передбачає скорочення часу виконання операцій, використання ресурсів системи та загальну продуктивність програмного продукту.

1.2 Загальний огляд процесу рефакторингу

Процес рефакторингу охоплює кілька етапів, кожен з яких має свої особливості:

1. Аналіз існуючого коду - це перший етап, що включає в себе детальний огляд існуючого коду з метою виявлення потенційних проблем або функцій, які можуть бути оптимізовані.

2. Визначення функцій рефакторингу: аналіз дозволяє визначити певні частини коду, які потребують рефакторингу.

3. Планування рефакторингу: розробка детального плану дій, який описує способи вирішення кожної проблеми.

4. Впровадження змін: послідовне впровадження запланованих змін за допомогою методів рефакторингу.

5. Тестування та перевірка: щоб переконатися, що функціональність програми не постраждає, кожен етап рефакторингу проходить детальне тестування.

1.3 Роль рефакторингу в розробці програмного забезпечення

Рефакторинг є важливою частиною процесу розробки програмного забезпечення, оскільки він має вирішальне значення для підтримки та покращення якості існуючого коду. Це не просто внесення невеликих змін або виправлення помилок. Рефакторинг — це процес перегляду, аналізу та переписування коду з метою оптимізації, зменшення складності, покращення читабельності та зрозумілості.

Метою рефакторингу є створення основи, яка дозволить ефективно впроваджувати нові функції та технології. Це означає, що рефакторинг допомагає підготувати код для адаптації до нових змін, які можуть виникнути через бізнес-вимоги, зміни в технологіях або нові стандарти розробки.

Рефакторинг є важливим інструментом для створення актуальної, гнучкої та легко масштабованої архітектури програми. Використовуючи його, можна гарантувати, що архітектура програми залишається незмінною, структурованою та готовою до розширення, якщо це буде потрібно.

1.4 Забезпечення сталого розвитку проекту

Рефакторинг надає розробникам нові можливості, дозволяючи їм адаптуватися до змін у технологіях або вимогах, не вимагаючи повного перебудовування проекту з нуля. Оскільки переписування коду не вимагає значних ресурсів і часу, це значно зменшує витрати на розробку та підтримку програмного забезпечення.

З іншого боку, даний процес забезпечує постійне вдосконалення та оптимізацію програмного забезпечення, а також зменшує витрати. Це означає, що код постійно вдосконалюється, що підвищує продуктивність, ефективність і надійність програми.

Рефакторинг дозволяє програмному забезпеченню залишатися актуальним і відповідати сучасним стандартам і технологіям. Це дуже важливо, оскільки технології розвиваються швидко, і програмне забезпечення, яке не адаптується до цих змін, може стати застарілим швидко.

1.5 Підвищення продуктивності команди

Ефективна розробка програмного забезпечення вимагає чистого та добре структурованого коду. Він не тільки підвищує продуктивність розробки, але й полегшує швидке виявлення та виправлення помилок. Це означає, що команди можуть швидше впроваджувати нові функції та працювати разом легше, що призводить до кращої якості продукту та меншого часу його випуску на ринок.

Процес перегляду та вдосконалення існуючого коду, робить інтеграцію нових розробників у проект більш простішою. Він покращує

розуміння коду новими членами команди, що знижує ймовірність помилок і сприяє більш ефективній роботі команди.

Рефакторинг також сприяє кращому розумінню коду кожним членом команди. Це означає, що кожен член команди може сприяти проекту, знаючи, як він працює, і як його можна покращити.

1.6 Забезпечення високої якості програмного продукту

Рефакторинг програмного забезпечення — це важливий процес, який гарантує, що програмне забезпечення залишається надійним, ефективним і легко адаптується до нових вимог ринку. Це означає, що програмне забезпечення може швидко та ефективно адаптуватися до змін в бізнес-середовищі, технологіях або вимогах користувачів.

Висока якість коду є ще одним важливим компонентом, який сприяє покращенню продуктивності програм. Програми функціонують ефективно, надійно та стабільно, коли їхній код чистий, добре організований і добре документований. Крім того, це гарантує, що код легко читати, змінювати та підтримувати розробниками, що скорочує час, потрібний для виявлення та виправлення помилок.

Це також забезпечує, що код може бути легко зрозумілим, модифікованим та підтримуваним розробниками, що зменшує час, необхідний для виявлення та виправлення помилок.

Розділ 2. Аналіз попереднього рішення (CMS Wordpress)

2.1 Огляд CMS Wordpress

WordPress — це відкрита система управління контентом (CMS), яка дозволяє створювати та керувати веб-сайтами. Вона була розроблена для забезпечення гнучкості для розробників і простоти використання для людей без технічних знань. Її основні переваги включають гнучкість,

масштабованість і адаптивність. Завдяки цій системі можна створювати веб-сайти різного розміру та складності, від звичайних блогів до великих корпоративних порталів і електронних магазинів. Оскільки WordPress підтримує оптимізацію пошукових систем (SEO), веб-сайти можуть підвищувати рейтинги в пошукових системах і залучати більше відвідувачів.

Унікальність WordPress полягає в його величезній спільноті користувачів і розробників. Ця спільнота створює та підтримує тисячі тем і плагінів, які дозволяють користувачам додавати нові функції та змінювати вигляд своїх веб-сайтів. Що стосується безпеки, WordPress відомий тим, що є надійним. Він регулярно оновлюється, щоб вирішити потенційні проблеми безпеки, і має активну спільноту, яка допомагає виявляти та виправляти помилки. Крім того, WordPress надає вбудовані засоби для резервного копіювання та відновлення даних, що допомагає запобігти втраті даних. Таким чином, WordPress є ефективним інструментом для створення та управління веб-сайтами, який підходить як початківцям, так і досвідченим розробникам. Багато різних типів веб-сайтів можна побудувати за допомогою WordPress, таких як блоги, корпоративні сайти, портфоліо, фотогалереї, онлайн-магазини, портали новин, освітні ресурси та багато іншого. Багато тем і плагінів для WordPress дозволяють це зробити. Користувачі можуть змінювати дизайн своїх веб-сайтів за допомогою тем, а плагіни додають нові функції або розширюють існуючі. WordPress має спільноту, яка є однією з його основних опор. Вона складається з мільйонів розробників і користувачів по всьому світу, які співпрацюють, обмінюються інформацією та ресурсами, і створюють нові плагіни та теми.

Крім того, ця спільнота активно співпрацює з командою розробників WordPress, щоб знайти та виправити помилки, покращити систему та додати нові функції. WordPress добре відомий тим, що він гнучкий. Використовуючи плагіни, він дозволяє користувачам легко налаштовувати свої веб-сайти, додавати новий контент, змінювати дизайн і додавати

додаткові функції. Крім того, WordPress дозволяє користувачам створювати мультимедійний контент, підтримуючи різні види медіа, такі як текст, зображення, аудіо та відео.

2.2 Архітектура та принципи роботи

WordPress, який базується на PHP, використовує MySQL для зберігання даних. Це означає, що він здатний працювати на більшості веб-серверів, які підтримують PHP та MySQL, що робить його сумісним з великою кількістю хостингових провайдерів. WordPress Theme System дозволяє розробникам і користувачам легко змінювати дизайн свого сайту без впливу на його функціональність. Це дозволяє змінити дизайн свого веб-сайту без ризику пошкодити плагіни або спричинити інші проблеми.

WordPress також має плагін «Архітектура», який дозволяє використовувати програми третіх сторін, щоб покращити функціональність вашого веб-сайту. Це означає, що встановлення плагінів, розроблених іншими розробниками, дозволяє додавати нові функції на ваш сайт. Це може бути будь-що, від форм зворотного зв’язку до інструментів SEO.

2.3 Переваги використання Wordpress для фотогалереї:

- Легкість використання: WordPress дозволяє користувачам без технічних навичок керувати контентом у фотогалереї завдяки зручному та простому інтерфейсу.

- Гнучкість: велика кількість тем і плагінів WordPress надає майже безмежні можливості для кастомізації веб-сайту, включаючи різні способи представлення зображень, інтеграцію з соціальними мережами та оптимізацію пошукової оптимізації.

- Спільнота та підтримка: велика кількість користувачів та розробників постійно розширює можливості WordPress, пропонуючи нові теми, плагіни та фікси поточних проблем.

2.4 Недоліки та обмеження Wordpress у контексті ресурсу фотогалереї:

- Продуктивність: швидкість завантаження сторінок фотогалереї може бути повільною через велику кількість плагінів, особливо з великою кількістю зображень.
- Безпека: популярність WordPress приваблює хакерів. У веб-сайту можуть виникнути вразливості, якщо розробники не надають достатньо уваги оновленням і безпеці плагінів.
- Стандартизація: хоча пластичність є однією з найважливіших переваг WordPress, вона також може бути недоліком, оскільки надмірна кастомізація може зробити оновлення ядра системи або плагінів, а також перехід на інші теми або розширення більш складним завданням.
- Оптимізація під мобільні пристрої: не всі теми та плагіни розроблені таким чином, щоб добре працювати на мобільних пристроях, що може створити проблеми для відвідувачів сайту, які використовують смартфони та планшети.
- Залежність: використання тем і плагінів від спільноти може призвести до залежності від розробників щодо підтримки та оновлень. У разі припинення підтримки або оновлень сайт може залишатися вразливим або втратити частину функціональності.

Розділ 3. Огляд технології JavaScript та Google API

3.1 Вступ до JavaScript

JavaScript — це мова програмування, яка була розроблена для використання в браузерах. Коли вона була розроблена Netscape у 1995 році, вона стала однією з найважливіших технологій веб-розробки. JavaScript передбачає виконання на стороні клієнта, тобто весь код виконується в браузері користувача замість сервера. Це дозволяє створювати веб-сайти, які реагують на дії користувачів у режимі реального часу, не перезавантажуючи сторінку. Подієва модель JavaScript є однією з його основних характеристик. Вона дозволяє швидко реагувати на дії користувача, такі як кліки миші, натискання клавіш та інші події. Це робить JavaScript ідеальним для створення інтерактивних веб-сайтів, які можуть змінюватися залежно від поведінки користувача без перезавантаження сторінки. Крім того, JavaScript підтримує AJAX, або асинхронний JavaScript та XML, технологію, яка дозволяє веб-сайтам взаємодіяти з сервером без перезавантаження сторінки.

Це означає, що JavaScript може надсилати запити на сервер, отримувати інформацію та оновлювати вміст сторінки без перезавантаження сторінки повністю. Це дозволяє створювати швидкі веб-сайти, незалежно від швидкості інтернет-з’єднання. Анімація на веб-сайтах, включаючи анімацію елементів інтерфейсу, також створюється за допомогою JavaScript. Це може включати візуальні ефекти, такі як рух елементів по сторінці, зміну кольорів і розмірів, серед іншого. Це підвищує інтерактивність веб-сайтів і дозволяє створювати більш привабливий і захоплюючий досвід користувача. Крім того, форма на стороні клієнта валідується за допомогою JavaScript. Це означає, що JavaScript може перевірити правильність даних користувача після введення їх у форму на веб-сайті.

Це може покращити загальний користувацький досвід, оскільки запобігає відправці неправильних або неповних даних. У результаті всіх цих функцій JavaScript є незамінним інструментом для сучасної веб-розробки.Використовуючи його, розробники можуть створювати веб-сайти, які не тільки добре виглядають, але й працюють ефективно,

надаючи користувачам гладкий і інтерактивний досвід. JavaScript все ще є однією з найпопулярніших мов програмування для веб-розробки завдяки своїй потужності та гнучкості. JavaScript, як мова програмування високого рівня, має багато вбудованих функцій і структур даних, що полегшує розробникам написання складного коду. Вона підтримує об'єктно-орієнтоване програмування, що дозволяє розробникам розробляти складні програми, які використовують класи та об'єкти. Крім того, JavaScript підтримує функціональне програмування, що дозволяє використовувати функції як об'єкти першого класу. Це дозволяє розробникам писати код, який є більш модульним і повторно використовуваним. Крім того, JavaScript підтримує асинхронне програмування, що означає, що він може виконувати завдання в фоновому режимі, не перешкоджаючи основному потоку виконання.

Це дозволяє створювати веб-сайти, які можуть виконувати складні обчислення або завантажувати дані з сервера без блокування GUI. Крім того, JavaScript підтримує обробку помилок, що дозволяє розробникам знаходити та обробляти помилки в коді. Це дозволяє створювати веб-сайти, які забезпечують кращий досвід користувачів, виправляючи помилки в реальному часі.

3.2 Основні можливості та переваги

Можливості:

- Інтерактивність і динамічний контент: вміст сторінок оновлюється в реальному часі, не вимагаючи повного перезавантаження комп'ютера.

- Асинхронні запити: використання AJAX і Fetch API для асинхронного зв'язку з сервером, що дозволяє зменшити навантаження на сервер і покращити час завантаження сторінок.

- DOM: Швидке та ефективне керування елементами сторінки дозволяє змінювати структуру та стиль веб-сайту.

Переваги:

- Універсальність: JS можна використовувати як на клієнтській стороні для браузерів, так і на сервері, за допомогою Node.js, що дозволяє повністю інтегрувати технології веб-розробки.

- Широка підтримка: JavaScript є універсальною мовою для розробки веб-додатків, оскільки його підтримують усі сучасні веб-браузери.

- Інтеграція: легко взаємодіє з іншими веб-технологіями та API, включаючи Google API, щоб розширити можливості веб-додатків.

3.3 Сучасні фреймворки та бібліотеки

- React: Ця бібліотека, розроблена Facebook, використовує компонентний підхід для побудови користувацьких інтерфейсів. React ідеально підходить для рефакторингу фотогалереї, оскільки він дозволяє легко управляти інтерфейсом, включаючи зображення, альбоми та навігацію. Це забезпечує гнучкість та ефективність при розробці користувацьких інтерфейсів.

-Angular: Цей потужний фреймворк, розроблений Google, надає комплексний підхід до створення односторінкових застосунків (SPA). Він містить вбудовані рішення для форм, асинхронної взаємодії з сервером, маршрутизації та багато іншого. Angular ідеально підходить для розробки великих, складних веб-додатків, таких як фотогалерея, які потребують багатої взаємодії та інтеграції між компонентами.

- Vue: Цей легкий, але потужний фреймворк дозволяє швидко створювати веб-додатки та забезпечує гнучкість. Vue підтримує компонентний підхід і двосторонній зв’язок даних, але він потребує менше коду порівняно з Angular. Vue є чудовим вибором для розробників фотогалереї, які цінують гнучкість і швидкість розробки, завдяки своїй простій UI та API.

3.4 Зв'язок з рефакторингом фотогалереї

Перехід від CMS WordPress до рішень, заснованих на JavaScript та сучасних фреймворків, таких як React, Angular і Vue, створює нові можливості для створення фотогалереї. Ці технології дозволяють створювати більш швидкі, гнучкі та інтерактивні веб-додатки.

- Інтерактивність та користувацький досвід: веб-додатки, які використовують JavaScript з фреймворками, мають більш інтерактивну взаємодію з фотогалереєю, яка дозволяє користувачам переглядати альбоми та миттєво завантажувати зображення.

- Масштабованість та продуктивність: використання модульної архітектури JavaScript-додатків замість монолітної структури WordPress підвищує масштабованість і продуктивність, дозволяючи ефективно обробляти велику кількість медіаданих і користувацьких запитів.

- Сучасні API та інтеграція: JavaScript полегшує інтеграцію з багатьма зовнішніми API та веб-сервісами, включаючи Google API, який надає більше можливостей для управління даними, аналізу контенту та роботи з зображеннями.

- Контроль та налаштування: Розробники можуть значно більше контролювати функціональність і інтерфейс фотогалереї, створюючи власні рішення на основі JavaScript та використання сучасних фреймворків. Це дозволяє налаштовувати поведінку програми відповідно до потреб користувачів і цілей проекту. Розробники мають можливість створювати унікальні користувацькі елементи, змінювати стилі та анімації, а також додавати нові технології, такі як машинне навчання та штучний інтелект, щоб розширити можливості галереї.

- Безпека та надійність: перехід на JavaScript та фреймворки, такі як React, Angular або Vue, гарантує, що веб-додатки захищені. Сучасні конструкції регулярно оновлюються, удосконалюючи вразливості та покращуючи засоби захисту. Використання сучасних протоколів аутентифікації та шифрування даних, таких як HTTPS, допомагає захистити дані користувачів і гарантує безпечний обмін даними між клієнтом і сервером.

- Адаптивність та доступність:з використанням JavaScript можна створювати адаптивні веб-додатки, які працюють на різних пристроях, таких як планшети, мобільні телефони та десктопи. Фреймворки дозволяють людям з обмеженими можливостями користуватися фотогалереєю без перешкод.

3.5 Огляд доступних Google API для роботи з зображеннями та медіа.

Google пропонує ряд API, які можуть бути інтегровані у веб-додатки для розширення їх функціональності, зокрема у контексті фотогалерей.

- Google Photos API: за допомогою цього API користувачі можуть легко та швидко інтегрувати свої зображення та альбоми з аккаунта Google Photo у свої додатки. Це дозволяє завантажувати нові зображення, створювати альбоми та керувати існуючими фотоальбомами.

- Google Drive API дозволяє користувачам переглядати файли та папки на Google Диску, включаючи фотографії. Це може бути використано для зберігання значних обсягів медіаданих у хмарі, що дозволяє легко отримати до них з будь-якого пристрою.

- Аналіз зображень можна покращити за допомогою Google Cloud Vision API, який дозволяє розпізнавати об’єкти, текст і логотипи, а також виявити небажаний контент. Це може бути використано для покращення пошукових можливостей фотогалереї та для автоматизації процесу тегування зображень.

3.6 Переваги використання Google API:

- Ефективне управління даними: Інтеграція Google Photos та Google Drive API, надає доступ до нових можливостей для ефективного управління великими кількостями медіафайлів. За допомогою цих API можна значно полегшити зберігання, доступ та організацію фотографій.

- Покращення якості зображень: Використання Google Cloud Vision API для аналізу зображень дозволяє автоматично вдосконалювати якість фотографій, розпізнавати та виправляти поширені проблеми зображень, такі як розмитість або неправильна експозиція.

- Розширені можливості пошуку: Аналіз зображень за допомогою Cloud Vision API надає можливість реалізувати потужні пошукові функції у фотогалереї, дозволяючи користувачам шукати фотографії за вмістом, об'єктами на них або навіть за текстом.

3.7 Обмеження Google API:

- Обмеження та квоти: Щоб використовувати Google API ефективно, важливо стежити за лімітами запитів, оскільки перевищення їх може призвести до додаткових витрат або навіть припинення роботи сервісу. Розробники повинні планувати використання API з урахуванням обмежень, оптимізуючи запити та кешуючи дані, коли це можливо, щоб зменшити кількість запитів, надісланих серверами Google.

- Залежність від зовнішнього сервісу: Інтеграція з Google API вносить ступінь залежності від стороннього провайдера, що може призвести до проблем, таких як зміни політики використання, тарифів або навіть припинення роботи API. Розробники повинні бути готові швидко адаптувати свої проекти до можливих змін.

- Вимоги до безпеки та конфіденційності: Розробники повинні дотримуватися строгих правил безпеки та конфіденційності, коли працюють з даними користувачів через Google API, особливо коли обробляються фотографії. Це включає захист повідомлень за допомогою шифрування, безпечне зберігання токенів і дотримання правил, таких як GDPR.

- Технічна складність: Розробники повинні розуміти основи асинхронного програмування, аутентифікації OAuth 2.0, використання RESTful API та обробки JSON-відповідей, щоб ефективно працювати з Google API. Це вимагає певного рівня технічної компетенції та може потребувати додаткового часу на навчання та експерименти.

Розділ 4. Порівняльний аналіз Wordpress та JS + Google API

4.1 Технічні аспекти порівняння

WordPress базується на PHP та використовує MySQL для зберігання даних. Його архітектура ідеально підходить для використання готових тем і плагінів для створення блогів, корпоративних сайтів і онлайн-магазинів.

Але розширення та налаштування функціоналу може вимагати знань PHP і веб-розробки.

JavaScript і Google API покращують середовище розробки, дозволяючи створювати повністю персоналізовані веб-додатки з використанням сучасних технологій і фреймворків. З можливістю інтеграції серверної логіки за допомогою API розробка зосереджена на клієнтській частині.

4.2 Продуктивність та швидкість роботи

WordPress може працювати гірше через велику кількість плагінів і складну тему, що збільшує час завантаження сторінок. Тим часом оптимізація, яка включає мініфікацію ресурсів і кешування, може призвести до підвищення швидкості.

JavaScript та Google API покращують продуктивність односторінкових застосунків (SPA), які можуть динамічно завантажувати контент без перезавантаження сторінки. Відгуки інтерфейсу можна отримати миттєво за допомогою асинхронних запитів і сучасних методів оптимізації.

4.3 Гнучкість та масштабованість

Завдяки великій кількості тем і плагінів WordPress є однією з найпопулярніших платформ для створення веб-сайтів, яка пропонує користувачам широкий вибір дизайну та функцій. Це робить WordPress чудовим вибором для веб-дизайнерів, які хочуть створити унікальний веб-сайт за короткий проміжок часу. Але розгортання масштабних проектів на WordPress може вимагати значних ресурсів і енергії.

Це особливо актуально для проектів, які вимагають високого рівня продуктивності, а також для проектів, які вимагають кастомізації. Великі проекти можуть вимагати більшої кількості серверних ресурсів і часу на налаштування та оптимізацію.З іншого боку, JavaScript і Google API дозволяють створювати масштабовані веб-додатки, які мають складну логіку на стороні клієнта та сервера. Це дозволяє вам створювати потужні веб-додатки, які можуть виконувати складні обчислення та обробляти великі кількості інформації.

Розподіл і масштабування навантаження можна спростити за допомогою хмарних платформ і мікросервісної архітектури. Ваші

програми можна легко масштабувати за допомогою інфраструктури, запропонованої хмарними платформами, такими як Google Cloud або Amazon Web Services. Однак мікросервісна архітектура дозволяє розділити ваш додаток на менші, незалежні сервіси, які можуть бути масштабовані та розгорнуті окремо.

4.4 Вартість впровадження та підтримки

Завдяки безкоштовним темам і плагінам, а також простоті установки та налаштування, WordPress пропонує низьку вартість входу. Проте для оптимізації продуктивності, високого рівня кастомізації або інтеграції специфічних функцій може знадобитися залучення фахівців, що може призвести до більших витрат. Збільшення кількості відвідувачів і ресурсів сайту також може призвести до зростання вартості хостингу.

JavaScript + Google API вимагає більших початкових інвестицій у час та розробку, особливо при створенні складних односторінкових застосунків або інтеграції з різноманітними API. Розробники повинні мати глибоке розуміння використання сучасних технологій і фреймворків. З іншого боку, завдяки ефективному використанню ресурсів і легкості масштабування у хмарних середовищах веб-додатки на базі JavaScript можуть забезпечити кращу продуктивність і нижчі витрати на підтримку в довгостроковій перспективі.

4.5 Вплив на SEO

WordPress традиційно вважається чудовим SEO-плагіном, як-от Yoast SEO, який оптимізує контент і структуру сайту для пошукових систем. Покращення індексації та ранжування досягається за допомогою оптимізованої структури посилань, карт сайту та автоматичної генерації мета-тегів.

З іншого боку, JavaScript і Google API можуть створювати проблеми для оптимізації пошукових систем через те, що динамічний вміст завантажується асинхронно і може не бути доступним для пошукових ботів одразу. Тим не менш, сучасні методи та технології, такі як серверний рендеринг (SSR) у фреймворках React типу Next.js, вирішують ці проблеми та забезпечують повну підтримку SEO для SPA.

4.6 Інтеграція з іншими сервісами

WordPress пропонує легку інтеграцію з багатьма зовнішніми сервісами та платформами через плагіни, але індивідуальна інтеграція або нестандартні вимоги можуть вимагати додаткової розробки.

JavaScript і Google API допомагають інтегруватися з багатьма API та сервісами, такими як хмарні платформи, соціальні мережі, аналітичні інструменти та інші. Це дозволяє розробникам створювати більш взаємопов’язані та інтерактивні веб-додатки, які можуть використовувати дані та функції з різних джерел у режимі реального часу.

Наприклад, інтеграція Google Maps для відображення геолокації фотографій, використання Google Analytics для аналізу поведінки користувачів на сайті або використання Firebase для створення повноцінних веб-додатків з бекендом у режимі реального часу.

4.7 Адаптивність та користувацький досвід

WordPress забезпечує базову адаптивність через теми, що підтримують мобільні пристрої, але створення унікального користувацького досвіду, який повністю адаптований до різних пристроїв, може вимагати додаткових зусиль.

JavaScript і Google API дозволяють розробникам створювати адаптивні інтерфейси, дозволяючи налаштовувати поведінку та

відображення веб-додатку на будь-якому пристрої. Використання сучасних CSS-фреймворків разом із реактивними JavaScript-фреймворками, такими як Vue або React, оптимізує користувацький досвід на мобільних, планшетних та десктопних пристроях.

4.8 Розширення та кастомізація

WordPress дозволяє легко розширювати функціонал через плагіни та віджети, але глибока кастомізація може вимагати знань PHP та розробки власних тем або плагінів.

JavaScript і Google API надають практично безмежні можливості для кастомізації та розширення функціональності веб-додатків. Розробники можуть створювати унікальні інтерфейси та взаємодії для користувачів, інтегрувати складну логіку обробки даних і взаємодіяти з різними веб-сервісами та API, щоб забезпечити багатофункціональний і гнучкий веб-додаток.

4.9 Використання Ant Design

Ant Design - це система дизайну, розроблена командою Ant User-Experience Design Team. Вона ґрунтується на чотирьох основних принципах дизайну: природність, визначеність, значущість та розвиток. Мета цієї системи - уніфікація специфікацій інтерфейсу користувача та зменшення надмірності та витрат на виробництво, що допомагає дизайнерам продуктів зосередитися на покращенні користувацького досвіду.

Використання дизайн-шаблонів на рівні підприємства може значно підвищити впевненість команди R&D, заощадити час на дизайн та підтримати системну консистентність, дозволяючи дизайнерам зосередитися на ключових завданнях. Дизайн-шаблони відповідають

цінностям дизайну Ant Design та надають загальні рішення для повторюваних проблем дизайну в продуктах підприємства.

Ant Design пропонує широкий спектр компонентів для користувацького інтерфейсу для покращення вашого веб-додатку, постійно покращуючи користувацький досвід. Крім того, вони рекомендують деякі великі сторонні бібліотеки. React використовується для інкапсуляції бібліотеки компонентів, що втілюють концепції їхнього дизайну. Їхнім бажанням є залучення спільноти до впровадження їхньої системи дизайну в інші фронт-енд фреймворки за їхнім вибором. Ant Design широко використовується для створення веб-сайтів на рівні підприємства як в країні, так і за кордоном.

Використання: Ant Design застосовується для створення веб-додатків на рівні підприємства, включаючи набір високоякісних компонентів React, які допомагають розробникам створювати багаті, інтерактивні інтерфейси користувача.

Особливості: Ant Design використовує технологію CSS-in-JS для надання динамічної та змішаної можливості тем. Також він використовує рішення на рівні компонентів CSS-in-JS, що забезпечує кращу продуктивність вашого додатку.

Підтримка: Ant Design підтримує багато сучасних браузерів, включаючи Edge, Firefox, Chrome, Safari та Opera. Він також підтримує рендеринг на стороні сервера та Electron.

Переваги: Ant Design допомагає розробникам підвищити продуктивність та ефективність комунікації. Він має компонент, шаблон або значок для вирішення будь-якої проблеми дизайну. Ant Design також пропонує обширну бібліотеку для створення нативних крос-платформених додатків.

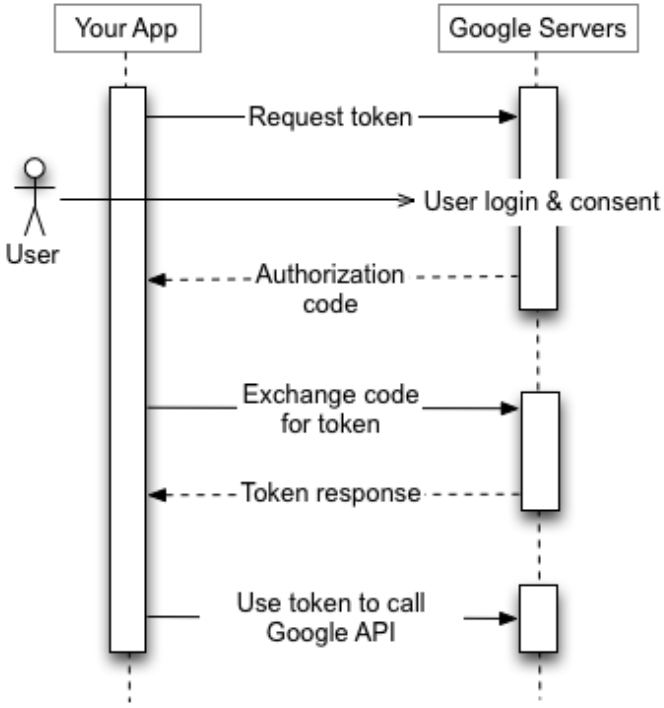
Інтеграція: Ant Design легко інтегрується з іншими популярними інструментами, такими як React, Bootstrap, jQuery UI, Ant Design Vue та refine.

4.10 Аутентифікація OAuth 2.0

Для того, щоб на сайті завантажувались зображення з бібліотеки Google Photos, потрібно виконати авторизацію у обліковий запис, з якого саме потрібно брати альбоми та фотографії. Для цього використовується протокол OAuth 2.0. Коли застосунок використовує цей протокол для авторизації, він діє від імені користувача, запитуючи токен доступу для отримання доступу до ресурсу, який застосунок ідентифікує за одним або кількома рядками області. Зазвичай користувача просять схвалити доступ.

Користувачі можуть обмінюватися певними даними з додатками за допомогою OAuth 2.0, зберігаючи конфіденційну інформацію, таку як імена користувачів, паролі та інші дані.

Сам процес авторизації та передачі токену користувача показано на



малюнку 1.1.

Малюнок 1.1 (модель аутентифікації)

Розділ 5 Розробка

5.1 Google Cloud Platform

Для початку потрібно створити обліковий запис на Google Cloud Platform. Саме цей сервіс допоможе нам здійснити авторизацію через OAuth 2.0. Коли обліковий запис вже створено, нам потрібно створити свій проект, після чого знаходимо бібліотеку Photos Library API. У цій бібліотеці ми отримаємо дані авторизації (реквізити для входу). Для створення реквізитів саме для моєї фотогалереї потрібно буде вказати адресу сайту та IP-адресу сервера. Для початку потрібно створити обліковий запис на Google Cloud Platform. Саме цей сервіс допоможе нам здійснити авторизацію через OAuth 2.0. Коли обліковий запис вже створено, нам потрібно створити свій проект, після чого знаходимо бібліотеку Photos Library API. У цій бібліотеці ми отримаємо дані авторизації (реквізити для входу).

Потрібно зареєструвати джерела, з яких додатку дозволено доступ до API Google, таким чином потрібно вказати:

- Ім’я, щоб визначити ідентифікатор клієнта.
- У полі "авторизовані джерела JavaScript" введіть джерело програми. У цьому полі не можна використовувати підставні знаки. Можно вказати кілька джерел, щоб додаток міг працювати на різних доменах, протоколах або піддоменах. URL-адреси, які ви вводите, можуть запускати запит OAuth.

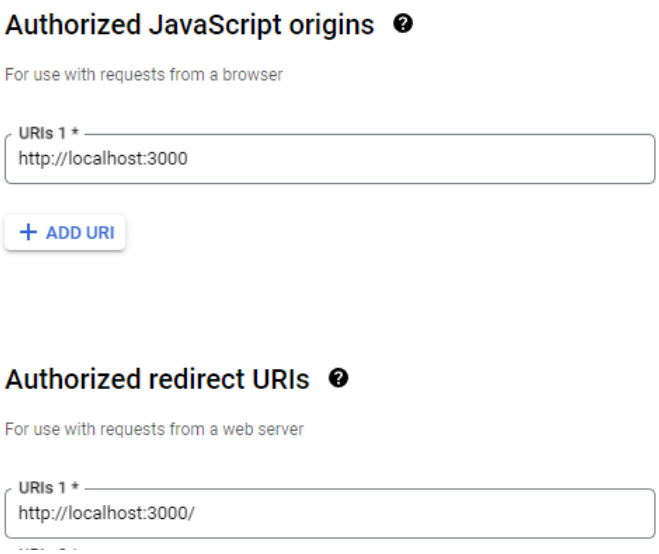
```
http://localhost:8080
https://myproductionurl.example.com
```

(у прикладі показано локальну URL-адресу).

Малюнок 5.1

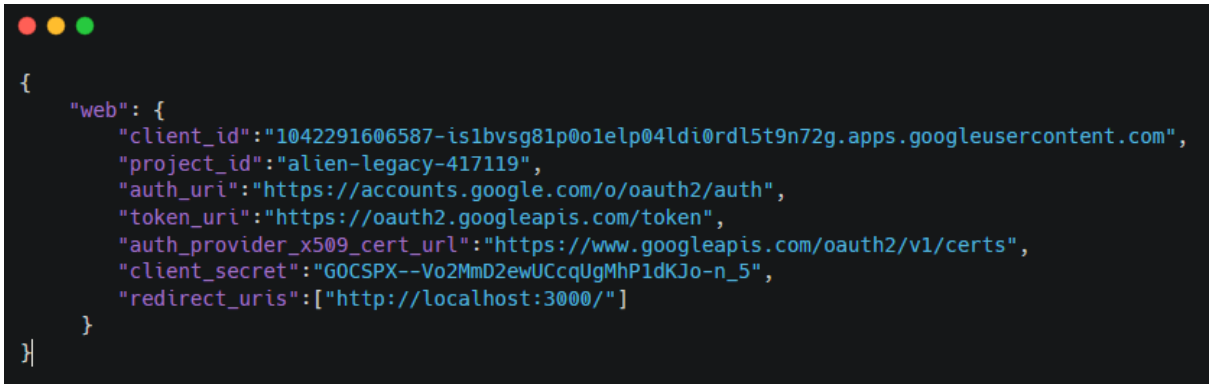
- Поле перенаправлення авторизованого URL - це кінцева точка, яка отримує відповіді від сервера OAuth 2.0. Кінцева точка, включає в себе ваше середовище розробки і вказує шлях у додатку.

Для створення реквізитів саме для моєї фотогалереї потрібно буде вказати адресу сайту та IP-адресу сервера.



Малюнок 5.2

Після введення усіх даних, ми зможемо отримати доступ до файлу, у якому знаходяться такі дані, як: ID клієнта, Секрет клієнта. Додаток може отримати доступ до API Google, використовуючи ці значення. Дані, які отримаємо, не можна публікувати, вони є конфіденційними. За допомогою цих даних можуть отримати доступ до ваших файлів.



Малюнок 5.3 Дані авторизації

5.2 Авторизація

Тепер додаємо кнопку авторизації на веб-сайт, завдяки якій ми можемо отримати токен авторизації. Цей токен надасть нам доступ до усіх наших альбомів. Однак автоматична авторизація без підтвердження у спливаючому вікні неможлива через налаштування конфіденційності Google. Щоб отримати токен, необхідне підтвердження користувача, і токен також оновлюється кожні півгодини. Навіть якщо ми отримали

токен доступу один раз, його не можна інтегрувати в код для автоматичного отримання доступу до усіх альбомів.

```
const authConfig = {
    client_id: "1042291606587-is1bvsg81p0o1elp04ldi0rdl5t9n72g.apps.googleusercontent.com",
    client_secret: "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",
    redirect_uris: ["http://localhost:3000/"],
    token_uri: "https://oauth2.googleapis.com/token"
};

ReactDOM.render(
    <React.StrictMode>
        <App authConfig={authConfig} />
    </React.StrictMode>,
    document.getElementById('root')
);
```

Малюнок 5.4

5.3 Дослідження фотогалерей

Після завершення створення методу авторизації для доступу до альбомів та фотографій, я розпочав аналіз існуючих публічних галерей. Метою було не лише визначення функціоналу, який необхідно надати користувачам, а й вибір оптимальної бібліотеки для створення дизайну. Я звернув увагу на можливості додавання функцій для редагування та пошуку фотографій прямо на веб-сайті. У своєму аналізі я ретельно розглядав такі ключові критерії:

1. Дизайн: Основною метою при створенні дизайну було забезпечення його зручності та лаконічності для користувачів. Наприклад, розташування більше ніж 4 альбомів в одному ряду може призвести до перевантаження сторінки та утруднення користування. Також, використання надмірно яскравих кольорів може негативно вплинути на сприйняття інтерфейсу. Тому при створенні дизайну було важливо дотримуватися принципів мінімалізму та зручності.

2. Якість зображень: Велика увага була приділена якості зображень у галереях. Деякі платформи зменшують якість зображень, щоб забезпечити швидке завантаження та зменшити використання ресурсів. Однак, це

рішення може бути необґрунтованим у випадку галерей з невеликою кількістю альбомів, де якість зображень є важливою для користувачів.

3. Функціональність: Під час аналізу було визначено кілька ключових моментів, які повинні бути враховані при розробці. Наприклад, оптимізація швидкості завантаження сторінок з фотографіями є важливою для забезпечення зручного перегляду контенту. Також, було визначено необхідність реалізації можливостей пошуку та фільтрації фотографій за різними параметрами, а також забезпечення мобільної сумісності для зручного перегляду на різних пристроях.

Характеристика	www.pexels.com	pixabay.com	stock.adobe.com
Мінімалістичний дизайн	Інтерфейс у мінімалістичному стилі, вже дуже легко - інтуїтивно зрозуміти	На головній сторінці сайту можна відразу побачити пошук зображень, кнопку пошуку розташовано майже на весь екран. Це дуже незручно. Також на головній сторінці є вибір категорій зображень, але кнопки зроблені занадто маленькими, що робить їх менш зручними для використання.	Малозрозумілий та неінтуїтивний інтерфейс. На головній сторінці дуже багато різного контенту, легко збентежитися.

Якість зображень	Сайт використовує стиснення зображень, але також має функцію завантаження зображення у його оригінальному розмірі	Сайт не використовує стиснення зображень, через що іноді може тривати довго завантаження деяких зображень, які мають великий розмір. Проте, на сайті присутні функції завантаження зображення у різних якостях.	Всі фото у максимальній якості, проте при завантаженні неможливо вибрати масштаб, який саме бажаєш завантажити. Доступна лише максимальна якість.
Мобільна сумісність	Дуже зручно використовувати сайт з телефону	Використання сайту на телефоні набагато зручніше, ніж на ПК.	Дуже приємно користуватися на телефоні. Сайт швидко завантажується.

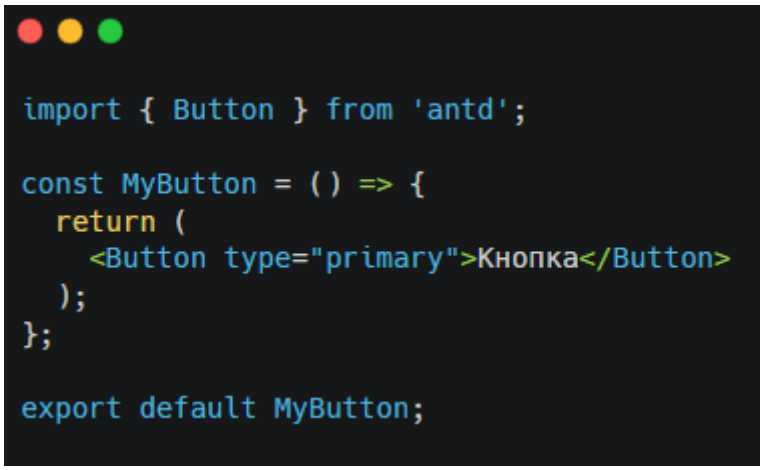
Таблиця 5.1

5.4 Імпорт бібліотеки Ant Design у проект

Після ретельного аналізу різноманітних публічних фотогалерей необхідно вибрати найкращу бібліотеку, яка відповідає вимогам проекту. Актуальність, швидкість, надійність і відповідність сучасним стандартам дизайну є основними критеріями. Ant Design 5.0 виявився найкращим варіантом після ретельної оцінки різних варіантів. Унікальний дизайн і підтримка мобільних пристроїв — це чудові функції цієї бібліотеки. Крім того, Ant Design 5.0 є дуже популярним серед розробників і вже тривалий час успішно продається на ринку. Таким чином, використання Ant Design 5.0 під час проектування допоможе забезпечити відповідність вимогам користувачів і забезпечити якісний та сучасний функціонал.

Малюнок 5.5 Імпорт бібліотеки

Основними складовими, до яких успішно застосовувалась ця бібліотека, були створення кнопок, організація структури веб-сайту, а також реалізація функціоналу управління навігацією між фотографіями.



Малюнок 5.6 Створення кнопки навігації

```
import { Collapse } from 'antd';

const { Panel } = Collapse;

const MyAlbum = () => {
  return (
    <Collapse defaultActiveKey={['1']}>
      <Panel header="Фотоальбом" key="1">
        Альбом
      </Panel>
    </Collapse>
  );
};

export default MyAlbum;
```

Малюнок 5.7 Створення анімації при відкритті альбому

5.5 Використання React Router DOM

React Router DOM — це набір інструментів для React, який дозволяє легко керувати шляхами та URL-адресами вашого додатку. Вона дозволяє створювати односторінкові програми (SPA), у яких відповідні частини відображаються відповідно до визначених маршрутів і змінювання URL не вимагає перезавантаження сторінки.

Основні компоненти React Router DOM, які було використано	
BrowserRouter	Цей компонент, який маршрутизується за допомогою HTML5 History API. Він створює контейнер, який відстежує зміни URL і показує необхідні

	компоненти.
Route	компонент, який визначає, який компонент має бути показаний за допомогою відповідної URL-адреси
Switch	Це компонент, який генерує перший дочірній маршрут або перенаправлення, який відповідає поточному URL. Щоб уникнути конфліктів, це дозволяє рендерити лише один маршрут.
Link	модуль, який, на відміну від тегів HTML, дозволяє переміщатися між сторінками. Він автоматично оновлює URL і відображає відповідний контент.
Redirect	функція, яка перенаправляє користувача на іншу веб-сторінку

Таблиця 5.2

```jsx
import React from 'react';
import { Link } from 'react-router-dom';

const NavigationMenu = () => {
  return (
    <nav>
      <ul>
        <li>
          <Link to="/">Головна</Link>
        </li>
        <li>
          <Link to="/contact">Контакти</Link>
        </li>
      </ul>
    </nav>
  );
}

export default NavigationMenu;
```

```jsx
import React from 'react';
import { Route, Switch } from 'react-router-dom';
import Home from './Home';
import About from './About';
import Contact from './Contact';

const Routes = () => {
  return (
    <Switch>
      <Route path="/about">
        <About />
      </Route>
      <Route path="/contact">
        <Contact />
      </Route>
      <Route path="/">
        <Home />
      </Route>
    </Switch>
  );
}

export default Routes;
```

Малюнок 5.8 Меню навігації

Малюнок 5.9 Визначення маршрутів та відображення компонентів

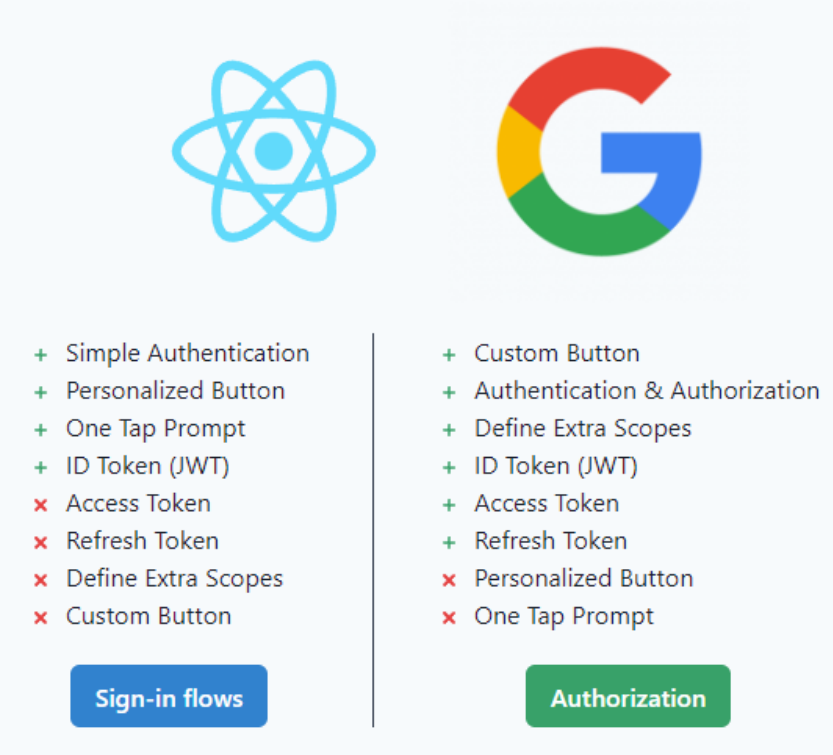
```jsx
import React from 'react';
import { BrowserRouter as Router } from 'react-router-dom';
import NavigationMenu from './NavigationMenu';
import Routes from './Routes';

const App = () => {
  return (
    <Router>
      <div>
        <NavigationMenu />
        <Routes />
      </div>
    </Router>
  );
}

export default App;
```

Малюнок 5.10 Головний компонент

5.6 Використання **React-Oauth/google**

React-oauth як ключова бібліотека для забезпечення безпечної авторизації на веб-сайтах. Завдяки цій бібліотеці, користувачі можуть легко та швидко увійти в свій аккаунт, натискаючи лише одну клавішу. Це рішення вражає порівняно з іншими бібліотеками React, де потрібно витратити багато часу на виконання декількох кроків для авторизації. Крім того, react-oauth/googl надає користувачам зручну функцію автоматичного оновлення токену, що гарантує безперервну роботу системи без необхідності ручного втручання. За допомогою цієї бібліотеки, веб-розробники можуть значно спростити процес авторизації на своїх сайтах,

забезпечуючи при цьому високий рівень безпеки та зручності для користувачів.



Малюнок 5.11 Порівняння звичайної бібліотеки React з React-Oauth

Завершуючи дипломну роботу на тему «Рефакторинг ресурсу фотогалереї з CMS WordPress на JS + Google API», я хочу описати основні результати та висновки. У процесі виконання роботи був проведений ретельний аналіз функціональних можливостей і обмежень існуючого ресурсу. Основними проблемами, з якими зіткнувся, були обмежені можливості управління контентом і низька швидкість веб-сайту, що негативно вплинуло на користувацький досвід. Крім того, обмеження політики використання Google Photos API викликали проблеми; це ускладнювало автоматичну авторизацію на веб-сайті та призвело до втрати токену через деякий час, що вимагало повторної авторизації на веб-сайті.

Вдалось вирішити ці більшість цих проблеми та впровадити низку покращень завдяки переходу від CMS WordPress на JavaScript + Google Photos API. Використання JavaScript покращило швидкість перегляду сторінок і зробило їх більш динамічними та інтерактивними. Можливості роботи з фотографіями покращилися завдяки інтеграції з Google API, яка забезпечує простий пошук, фільтрацію та відображення геоданих.

ДЖЕРЕЛА

1. React-Oauth - https://www.npmjs.com/package/@react-oauth/google

2. Ant design - https://ant.design/
3. OAuth 2.0 - https://developers.google.com/identity/protocols/oauth2?hl=ru
4. Java Script - https://developer.mozilla.org/en-US/docs/Web/JavaScript.
5. Google Photos api - https://developers.google.com/photos/library/guides/overview?hl=ru https://max-coding.medium.com/loading-photos-and-metadata-using-google-photos-api-with-python-7fb5bd8886ef
6. CMS WordPress - https://wordpress.org/documentation/
7. React Router - https://reactrouter.com/en/main/start/tutorial
8. JavaScript повне керівництво; довідник по найпопулярнішій мові програмування, 7-е изд. Фленаган Д.