ЗВІТ З ПЕРЕВІРКИ НА ПЛАГІАТ

ЦЕЙ ЗВІТ ЗАСВІДЧУЄ, ЩО ПРИКРПЛЕНА РОБОТА

Ротко_КІ-119К

БУЛА ПЕРЕВІРЕНА СЕРВІСОМ ДЛЯ ЗАПОБІГАННЯ ПЛАГІАТУ MY.PLAG.COM.UA І МАЄ:

СХОЖІСТЬ

4%

РИЗИК ПЛАГІАТУ

43%

ПЕРЕФРАЗУВАННЯ

1%

НЕПРАВИЛЬНІ ЦИТУВАННЯ

0%

Назва файлу: Ротко_КІ-119К.pdf
Файл перевірено: 2023-06-07
Звіт створено: 2023-06-07

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ЗАПОРІЗЬКИЙ ІНСТИТУТ
(library.econom.zp.ua) ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Предметно-циклова комісія інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНА

Голова ПЦК _____

спеціаліст в/к Сабанов
(library.econom.zp.ua) С.О.

ВИПУСКНА РОБОТА МОЛОДШОГО СПЕЦІАЛІСТА
(library.econom.zp.ua)

СТВОРЕННЯ ТЕМАТИЧНОГО ВЕБ-РЕСУРСУ НА БАЗІ WIKI-ТЕХНОЛОГІЙ

Виконав
ст. гр. ІПЗ-119К9 _____ Ротко К.Р.

Керівник
професор _____ Сабанов С.О.

Запоріжжя
2023

СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Предметно-циклова комісія (library.econom.zp.ua) інформаційних технологій

ЗАТВЕРДЖУЮ

Голова ПЦК спеціаліст в/к

Сабанов

(library.econom.zp.ua) С.О.

“17” січня 2022 року

ЗАВДАННЯ

НА ВИПУСКНУ РОБОТУ МОЛОДШОГО СПЕЦІАЛІСТА

студенту гр. (77.93.36.128) *КІ-119К9*

спеціальності: *121 – Інженерія програмного забезпечення*

(library.econom.zp.ua) Ротко Кирилу Романовичу

(прізвище, ім'я, по батькові)

1. Тема: «*Створення тематичного веб-ресурсу на базі Wiki-технологій*»

затверджена наказом по інституту: № _____від 04 березня 2022 року

2. Термін здачі студентом закінченої роботи: 18 червня 2022 року (library.econom.zp.ua)

3. Перелік питань, що підлягають розробці:

1. *Провести огляд літератури та інтернет-джерел, присвячених тематиці випускної роботи. (library.econom.zp.ua)*
2. *Виконати огляд найбільш популярних Wiki-двигунів та Wiki-платформ та навести основні характеристики.*
3. *Розглянути інструментальні засоби створення Wiki-ресурсів з метою вибору оптимальних для створення проекту.*
4. *Визначитися з основними функціональними блоками проекту та структурою бази даних.*
5. *Реалізувати алгоритм роботи додатку.*
6. *Протестувати розробку та провести остаточне налагодження*

7. *Створити достатнє наповнення БД для нормального користування системою.*
8. *Продумати питання хостингу*
9. *Оформити результати роботи у вигляді пояснювальної записки, що відповідає стандартам підприємства щодо оформлення випускних робіт.*

4. Календарний графік

№ етапу	Зміст	Термін виконання	Готовність (%), підпис керівника	Підпис керівника про повну готовність етапу, дата
1	Збір практичного матеріалу за темою	16.01.23-17.02.23		
2	І атестація. І розділ випускної роботи	27.03.23-01.04.23		
3	ІІ атестація. ІІ розділ випускної роботи	01.05.23-06.05.23		
4	ІІІ атестація. ІІІ розділ випускної роботи, висновки (library.econom.zp.ua) та рекомендації, додатки, реферат.	29.05.23-03.06.23		
5	Перевірка випускної роботи (library.econom.zp.ua) на оригінальність	15.05.23-12.06.23		
6	Доопрацювання випускної роботи, підготовка презентації, отримання відгуку керівника та рецензії	05.06.23-10.06.23		
7	Попередній захист випускної роботи	12.06.23-18.06.23		
8	Подача випускної роботи на кафедру	за 3 дні до захисту		
9	Захист випускної роботи	19.06.23-24.06.23		

Дата видачі завдання: 03 жовтня 2022 р.

Керівник випускної роботи_____ Завдання прийняв до виконання
(підпис)

_____ Сабанов С.О.
(підпис студента)

Ротко К.Р.

Випускна робота молодшого спеціаліста містить 46 сторінок, 20 рисунків, 1 додаток, 15 бібліографічних посилань.

Об’єкт розробки – тематичний веб-сайт.

Метою роботи є розробка (library.econom.zp.ua) енциклопедичного веб-ресурсу, що працює з використанням Wiki-технологій.

У випускній роботі розглядаються основні принципи та етапи побудови довідкових матеріалів на базі Wiki-двигунів та Wiki-платформ, а також інструментальні засоби створення подібних ресурсів.

Детально описано процес розробки тематичного веб-сайту, що містить інформаційно-довідкові матеріали щодо фентезійного світу, створеного за літературним циклом польського письменника Анджея Сапковського «Відьмак».

Додаток розроблено на базі клієнт-серверної архітектури та адаптований під різні пристрої: телефон, планшет та персональний комп’ютер.

BACKEND, FRONTEND, HTML, NODE.JS, REACT JS, SCSS,
WEB-САЙТ, WIKI, WIKI-ДВИГУН, WIKI-ПЛАТФОРМА, БАЗА ДАНИХ,
КЛІЄНТ-СЕРВЕР, СУБД

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Слово / словосполучення	Скорочення
C	
Content Management System	CMS
D	
Document Object Model	DOM
G	
General Public License	GNU (GPL)
H	
HyperText Markup Language	HTML
J	
JavaScript	JS
JavaScript XML	JSX
M	
My Structured Query Language	MySQL
P	
Personal Home Page	PHP (doi.org)
S	
Syntactically Awesome Style Sheets	SCSS
Search Engine Optimization	SEO
Secure Sockets Layer	SSL
U	
User Interface	UI
Uniform Resource Locator	URL
V	

Virtual Private Server	VPS
Visual Studio Code	VS Code
Б	
База даних	БД

ВСТУП

У світі сучасних технологій веб-ресурси займають все більше значуще місце у суспільному житті. Особливо це стосується такого популярного формату як веб-сайт-енциклопедія. Інтернет-енциклопедії з'явилися ще на початку 2000-х, але з того часу вони зазнали значних змін і тепер пропонують користувачам все більше інтерактивних можливостей. Чимала кількість таких ресурсів використовує досить специфічну технологію – вікі.

Технологія вікі (англ. Wiki technology) – це система управління веб-сайтом або інформаційною базою даних, яка дозволяє користувачам створювати, редагувати та видаляти інформацію на веб-сторінках за допомогою веб-браузера. Технологія вікі базується на створенні відкритих веб-сайтів, до яких будь-який користувач може мати доступ та вносити свої внески у вигляді тексту, зображень, відео, звукових файлів тощо.

На відміну від традиційних систем управління контентом (CMS), в яких зміни вносяться тільки авторизованими користувачами зі спеціальними правами доступу, вікі дозволяють будь-якому користувачу редагувати та модифікувати сторінки. Крім того, вікі є співпрацюючим середовищем, де різні користувачі можуть спільно працювати над створенням та вдосконаленням змісту.

Одним з найважливіших прикладів використання технології вікі є Вікіпедія – вільна енциклопедія, до якої будь-який користувач може додавати свої знання та досвід. Вікіпедія містить мільйони статей, які переглядаються мільярдами користувачів по всьому світу.

Технологія вікі може бути використана для створення різноманітних проектів, від відкритих енциклопедій до проектів зі збору знань, блогів, форумів, інтернет-магазинів та інших. Вікі дозволяють користувачам швидко та ефективно об'єднувати свої знання та досвід для створення чого-небудь цікавого та корисного.

У даній випускній роботі розглядається розробка тематичного веб-

сайту-енциклопедії з використанням wiki-технології. Сайт висвітлює тематику фентезійного світу, створеного за літературним циклом польського письменника Анджея Сапковського «Відьмак». Цей всесвіт є одним з найпопулярніших фентезійних світів останніх десятиліть. Головною метою проекту є створення високоякісного веб-ресурсу, який міститиме повну та систематизовану інформацію про всі аспекти всесвіту "Відьмака", від персонажів та їхніх характеристик до історії та культури цього унікального світу.

РОЗДІЛ 1
ТЕХНОЛОГІЯ WIKI ТА ЇЇ ОСНОВНІ ХАРАКТЕРИСТИКИ

В сучасному світі, де інформація є однією з найцінніших ресурсів, можливість швидкої та ефективної обробки та збереження даних стала надзвичайно важливою. В зв'язку з цим, технології, що дозволяють створювати та редагувати інформацію в реальному часі, отримали величезну популярність. Однією з таких технологій є вікі, що є потужним інструментом для колективної роботи з інформацією та знаннями.

У цьому розділі випускної роботи викладено відомості про технологію вікі та її основні характеристики. Оглянуто принцип роботи вікі-систем, основні можливості вікі-двигунів та їх різноманітність. Крім того, розглянуто популярні вікі-платформи та способи самостійного хостингу вікі-сайтів.

Цей розділ є важливою основою для розуміння технології вікі та дозволить дослідити можливості створення веб-сайту-енциклопедії на тему всесвіту "Відьмака".

1.1. Загальна концепція wiki та основні відомості про технологію

Wiki-технологія є однією з найпоширеніших технологій для колективного редагування веб-сторінок. Сам термін "wiki" походить від гавайського слова "wiki wiki", що означає "швидко" або "швидкий". Концепція wiki передбачає створення веб-сторінок, які можуть бути редаговані не лише їх авторами, але й будь-якими користувачами з доступом до Інтернету. Це дозволяє створювати колективні знання та спільноти, які можуть розвиватись та розширюватись з часом.

Основним принципом wiki є простота та доступність для всіх. Відсутність необхідності знати складні мови програмування дозволяє будь-

кому внести свій вклад в створення та редагування веб-сторінок. Крім того, технологія wiki передбачає інтерактивність та співпрацю, що дозволяє користувачам обговорювати та розвивати зміст сторінок.

Основним елементом wiki-технології є система маркерів або тегів, що дозволяє форматувати текст та визначати його роль на сторінці. Також, wiki передбачає можливість створення посилань між різними сторінками, що дозволяє створювати навігаційні структури та зручно переходити між сторінками з подібною тематикою.

Таким чином, wiki-технологія є потужним інструментом для створення колективних знань та співпраці в Інтернеті. Для створення веб-сайту-енциклопедії на тему Всесвіту "Відьмака" використання wiki-технології є доцільним, оскільки дозволяє залучити до співпраці широку аудиторію.

1.2. Wiki-двигуни

1.2.1. Призначення

Wiki-двигун (або wiki система) – це програмне забезпечення, яке дозволяє створювати та управляти веб-сайтом з відкритим змістом. Основна ідея wiki полягає у тому, що відвідувачі можуть додавати, редагувати та видаляти вміст сайту. Це дає можливість користувачам взаємодіяти з сайтом та доповнювати його знаннями та досвідом.

Основним призначенням wiki-двигунів є створення відкритих енциклопедій, баз даних знань та навчальних ресурсів, де користувачі можуть знайти та додати інформацію на різні теми.

1.2.2. Огляд найбільш популярних двигунів Wiki

Сьогодні існує багато різних wiki-двигунів, які використовуються для створення відкритих веб-сайтів. Найбільш популярними з них є такі.

MediaWiki – це безкоштовний Wiki-двигун, який був розроблений спеціально для Вікіпедії. Це серверне програмне забезпечення, яке розповсюджується на умовах GNU (GPL), тобто його відкритий код дозволяє користувачам змінювати та адаптувати програмне забезпечення під свої потреби. MediaWiki може використовуватися на групах серверів (server farm), щоб забезпечити безперебійне обслуговування множинних запитів. Двигун використовує класичну зв'язку PHP+MySQL, має велику потужність і дозволяє системі добре масштабуватися.

Користувачі, які не знайомі з технологіями створення веб-сторінок, можуть легко редагувати сторінки, використовується інтуітивно зрозумілий формат даних користувача – вікітекст. В процесі редагування сторінки вносяться до БД для запобіганню втрати попередніх даних, що дозволяє анулювати внесені зміни у критичних випадках. [1]

DokuWiki є легким Wiki-двигуном з відкритим кодом, який може бути встановлений на будь-якому сервері з підтримкою PHP. Цей двигун має простий та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко створювати, редагувати та публікувати вміст на своєму веб-сайті.

DokuWiki не вимагає бази даних, що робить його ідеальним для використання на простих веб-сайтах. Вміст зберігається в текстових файлах, що дозволяє зберігати його в репозиторії контролю версій та редагувати за допомогою звичайного текстового редактора.

Основні особливості DokuWiki:

- Простий інтерфейс користувача.
- Безкоштовний та з відкритим кодом.
- Легко і швидко встановлюється та налаштовується.
- Забезпечує безпечну аутентифікацію користувачів.
- Дозволяє створювати підкатегорії та сторінки з різними рівнями доступу.
- Підтримує різноманітні плагіни для розширення функціональності.

- Є велика база знань та документації для користувачів [2]

WikkaWiki – це безкоштовний Wiki-двигун з відкритим кодом, що розробляється спільнотою розробників. Цей двигун дозволяє легко створювати відкриті веб-сайти з метою спільного користування знаннями та інформацією.

Основні особливості WikkaWiki:

- Простий та зрозумілий інтерфейс користувача.
- Можливість використання віджетів та шаблонів для налаштування вигляду веб-сайту.
- Підтримка різних мов та кодувань.
- Можливість редагування сторінок без реєстрації користувача.
- Підтримка плагінів для розширення функціональності.
- Можливість використання віджетів та шаблонів для налаштування вигляду веб-сайту. [3]

Tiki Wiki CMS Groupware – це безкоштовний Wiki-двигун з відкритим кодом, який надає повний набір інструментів для співпраці та керування вмістом. Цей двигун може бути використаний для створення різних типів веб-сайтів, таких як інтернет-магазини, блоги, форуми тощо.

Основні особливості Tiki Wiki CMS Groupware:

- Безкоштовний та з відкритим кодом.
- Можливість редагування вмісту безпосередньо на веб-сторінці.
- Підтримка багатьох мов та кодувань.
- Наявність вбудованого чату та інструментів для спілкування.
- Можливість керування проектами та завданнями.
- Підтримка плагінів для розширення функціональності, таких як календар, форми, галереї тощо.
- Наявність інструментів для аналізу відвідуваності веб-сайту та відстеження статистики. [4]

Розглянуті Wiki-двигуни мають різні переваги та недоліки, і вибір

залежить від конкретних потреб користувача та специфіки проекту. Важливо враховувати такі параметри, як розмір проекту, кількість користувачів, потрібні функції та ресурси для розгортання веб-сайту.

1.3. Популярні Wiki-платформи

Wiki-платформа – це веб-програмне забезпечення, яке дозволяє користувачам створювати та редагувати веб-сторінки з відкритим змістом. На сьогоднішній день існує багато популярних Wiki-платформ, серед яких можна виділити такі:

Вікіпедія – це безкоштовна, вільна та відкрита інтернет-енциклопедія, яка містить мільйони статей на різні теми. Статті у Вікіпедії написані користувачами з усього світу, які можуть додавати, редагувати та видаляти інформацію. Платформа створена на базі вікі-технології та підтримується Вікімедіа Фундацією – некомерційною організацією, яка забезпечує безкоштовний доступ до знань та культурних ресурсів.

Вікіпедія має декілька мовних версій та доступна для користувачів з усього світу. Кожна стаття містить інформацію про певну тему, яка може бути використана для навчання, роботи та особистого розвитку. Вікіпедія також дозволяє користувачам вносити свій внесок у проект, вносячи нові статті або редагуючи існуючі. (рисунок 1.1)

Вікімедіа – це набір вікі-платформ, які включають Вікіпедію, Вікіцитатник, Вікіджерела, Вікідані та інші. Ці платформи побудовані на базі вікі-технології та підтримуються Вікімедіа Фундацією. Кожна з платформ має свою унікальну функціональність та спрямована на різні типи знань та даних.

Вікімедіа Фундація прагне забезпечити безкоштовний доступ до знань та культурних ресурсів для користувачів з усього світу. Усі платформи Вікімедіа забезпечують відкритий доступ до знань, що означає, що

користувачі можуть вносити свій внесок у створення та підтримку знань. Вікімедіа прагне зберігати та поширювати знання в інтересах загальної громадськості та сприяти розвитку освіти та науки.

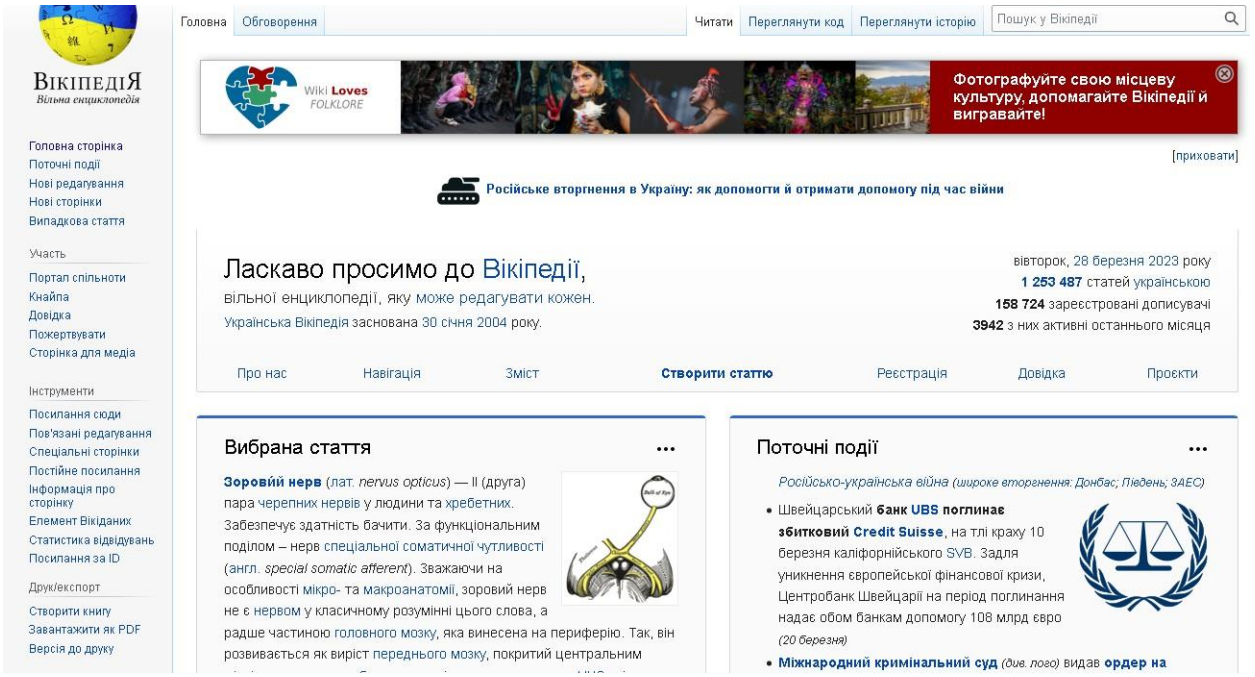

Рисунок 1.1 – Сайт Вікіпедія

Wikia – це платформа, яка дозволяє користувачам створювати власні вікі-сайти на будь-яку тему. Користувачі можуть створювати вікі-сайти на будь-яку тему, включаючи відеоігри, фільми, телешоу, музику та спорт. Wikia надає безкоштовний хостинг та можливість користувачам налаштовувати свої вікі-сайти за допомогою шаблонів та плагінів.

Користувачі можуть створювати сторінки, додавати контент та зображення, редагувати сторінки та взаємодіяти з іншими користувачами. Wikia дозволяє створювати спільноти навколо вікі-сайтів, які можуть включати форуми, чати та інші інтерактивні функції.

Fandom – це платформа, яка спеціалізується на веб-сайтах про відеоігри, фільми та телешоу. Користувачі можуть створювати власні вікі-сайти на будь-яку тему, пов'язану з розвагами. Fandom надає безкоштовний хостинг та можливість користувачам налаштовувати свої вікі-сайти за

допомогою шаблонів та плагінів.

Користувачі можуть створювати сторінки, додавати контент та зображення, редагувати сторінки та взаємодіяти з іншими користувачами. Fandom дозволяє створювати спільноти навколо вікі-сайтів, які можуть включати форуми, чати та інші інтерактивні функції.

Одним з головних принципів Fandom є підтримка спільноти, що заснована на співпраці та взаємодії користувачів. Fandom надає засоби для комунікації та співпраці між користувачами, включаючи форуми, чати та можливість створювати спільнотні блоги.

Крім того, Fandom співпрацює з багатьма видавцями та розробниками відеоігор, фільмів та телешоу, щоб забезпечити користувачам актуальну та точну інформацію про ці твори мистецтва. Fandom також пропонує спеціальні програми для користувачів, такі як програма "Fandom на кшталт Вікімедіа", яка дозволяє користувачам отримувати фінансову підтримку за внесок у створення та підтримку вмісту на платформі.

Загалом, Wikia та Fandom є платформами, які дозволяють користувачам створювати власні вікі-сайти та спільноти, взаємодіяти з іншими користувачами та вносити свій внесок у створення та підтримку знань.

1.4. Самостійний хостинг Wiki-сайту

Створення власного wiki-сайту з самостійним хостингом може бути досить складним процесом, але він надає більшу свободу в налаштуванні та розробці веб-сайту.

Основні кроки для створення wiki-сайту з самостійним хостингом наведено далі.

1) Вибір хостинг-провайдера. Спочатку потрібно вибрати хостинг-провайдера, який надає віртуальний сервер (VPS) або фізичний сервер для хостингу веб-сайту. Важливо звернути увагу на технічні

характеристики, які включають обсяг пам'яті, кількість ядер процесора, доступну пропускну здатність та інші важливі параметри. Крім того, треба звернути увагу на рівень підтримки, який надається хостинг-провайдером.

2) Вибір Wiki-двигуна. Після вибору хостинг-провайдера потрібно встановити wiki-двигун, який буде використовуватися для розробки веб-сайту. Є багато безкоштовних та комерційних wiki-двигунів, таких як MediaWiki, DokuWiki, TikiWiki та інші.
3) Налаштування веб-сервера. Для запуску wiki-двигуна на сервері потрібно налаштувати веб-сервер. Більшість wiki-двигунів підтримують веб-сервер Apache, але є інші варіанти, такі як Nginx та Lighttpd. Налаштування веб-сервера може бути складним процесом, тому рекомендується звернутися до документації wiki-двигуна та хостинг-провайдера для отримання допомоги.
4) Налаштування бази даних. Більшість wiki-двигунів використовують базу даних для збереження контенту веб-сайту. Перед початком використання wiki-двигуна потрібно налаштувати базу даних на вашому сервері. Зазвичай, використовуються реляційні бази даних, такі як MySQL, PostgreSQL або SQLite. Перед налаштуванням бази даних, важливо переконатися, що вона встановлена на сервері та що є необхідні права доступу до неї.
5) Установка Wiki-двигуна. Після того, як встановлено веб-сервер та налаштовано базу даних, потрібно встановити wiki-двигун. Для цього можна скористатися командною строкою або інсталятором, який зазвичай надається з wiki-двигуном. Необхідно також налаштувати параметри wiki-двигуна, які включають налаштування безпеки, доступу та ін.
6) Налаштування забезпечення безпеки. При створенні веб-сайту, особливо якщо він доступний в Інтернеті, необхідно враховувати проблеми безпеки. Необхідно буде налаштувати захист веб-сайту від

хакерських атак, зловмисних програм та інших загроз. Для цього можна використовувати різні методи, такі як налаштування файрвола, встановлення SSL-сертифіката, налаштування аутентифікації та авторизації, а також використання різних інструментів для виявлення та виправлення проблем з безпекою.

7) Налаштування резервного копіювання. Щоб захистити веб-сайт від втрати даних у разі збою апаратного забезпечення або іншої проблеми, необхідно налаштувати резервне копіювання. Можна використовувати різні інструменти для автоматичного резервного копіювання, такі як rsync, scp, tar та інші.
8) Налаштування SEO та аналітики. Щоб забезпечити успішне просування веб-сайту в пошукових системах та відстежувати його ефективність, необхідно налаштувати SEO та аналітику. Для цього необхідно виконати ряд дій, таких як додавання метатегів, налаштування файлу robots.txt, реєстрація веб-сайту в пошукових системах та використання аналітичних інструментів, таких як Google Analytics.

Після налаштування веб-сервера, бази даних, wiki-двигуна та забезпечення безпеки, веб-сайт повинен бути готовий до використання. Проте, технічне утримання веб-сайту – це постійний процес. Потрібно буде забезпечувати безпеку веб-сайту, виявляти та виправляти помилки, оновлювати програмне забезпечення та виконувати регулярні резервні копії, щоб захистити дані.

1.5. Висновки за першим розділом

У загальному, технологія Wiki дозволяє створювати веб-сайти з відкритим змістом, які можуть бути корисними для різних груп користувачів. Вона забезпечує співпрацю та обмін інформацією між користувачами, що

сприяє розвитку відкритої культури та знань.

Технологія Wiki є важливим інструментом для створення відкритих веб-сайтів, які дозволяють користувачам додавати та редагувати контент, а також анулювати зміни в разі випадків вандалізму або спаму.

Розглянуті платформи створення Wiki-ресурсів, зокрема MediaWiki, DokuWiki та WikkaWiki. мають свої переваги та недоліки, тому вибір конкретної залежить від потреб користувача та мети створення сайту. Хостинг-платформи зазвичай пропонують простий та зручний спосіб створення сайту, але можуть бути обмежені в можливостях налаштування та редагування сайту.

Альтернативним способом створення Wiki-сайту є самостійний хостинг. Він може бути складним для тих, хто не має достатньої технічної експертизи, але дозволяє повністю контролювати сайт і не залежати від інших компаній або платформ.

РОЗДІЛ 2
ІНСТРУМЕНТАЛЬНІ ЗАСОБИ СТВОРЕННЯ WIKI-РЕСУРСІВ

2.1. Основні компоненти Wiki-ресурсу

В даному розділі розглянемо основні компоненти Wiki-ресурсу, які використовуються для створення, редагування та відображення контенту. Загалом, такі ресурси мають ряд складових, які взаємодіють між собою, забезпечуючи зручний та швидкий доступ до інформації.

2.1.1. Frontend-засоби

Frontend-засоби це компоненти, що відповідають за зовнішній вигляд та інтерфейс взаємодії користувача з ресурсом. У проекті використовується Html, SCSS та React js для створення зручного та привабливого інтерфейсу.

Html [10, 14] – це мова розмітки, що використовується для створення структури сторінок. З його допомогою визначаються заголовки, абзаци, списки та інші елементи, що використовуються для відображення контенту.

SCSS – це мова стилів, що розширює звичайний CSS. [12, 15] Вона дозволяє використовувати змінні, функції та інші конструкції, що спрощують написання стилів та роблять їх більш зрозумілими.

React js – це бібліотека для створення інтерфейсів, яка дозволяє розбити сторінки на компоненти та забезпечує зручний механізм для їх взаємодії. Використовуючи React js, можна створювати перевикористовувані компоненти, що значно спрощує процес розробки та забезпечує однаковий вигляд різних елементів інтерфейсу.

2.1.2. Backend-підтримка

Другим важливим компонентом Wiki-ресурсу є backend-підтримка. Вона забезпечує зберігання та обробку даних, які зберігаються на сервері.

У проекті використовується Firebase як основний backend-провайдер. Firebase – це платформа розробки мобільних та веб-додатків, що надає різноманітні сервіси для розробки та підтримки додатків. Firebase дозволяє використовувати готові інструменти для створення серверного додатку без необхідності власноруч розробляти складні механізми збереження даних та їх синхронізації між клієнтами.

Firebase забезпечує такі сервіси, як база даних реального часу, зберігання файлів, аутентифікація та інші. База даних реального часу дозволяє зберігати (ir.nmu.org.ua) дані в режимі реального часу, що дозволяє користувачам побачити оновлення на веб-сторінці в режимі реального часу.

Firebase також має добре розроблений інтерфейс для роботи з базою даних та можливість використовувати її на різних мовах програмування, що дозволяє розробникам просто і швидко розробляти веб-додатки.

Використання Firebase в проекті значно зменшує час, необхідний для розробки серверної частини, що дозволяє більше уваги приділяти розробці клієнтської частини та функціоналу веб-додатку.

2.2. Бібліотека створення інтерфейсів користувача React

React – це відкрита JavaScript [13] бібліотека для створення інтерфейсів користувача (UI) на веб-сайтах або веб-додатках. За допомогою React, розробники можуть створювати динамічні та високопродуктивні веб-додатки, що забезпечують користувачам більш якісний досвід взаємодії з сайтом. [6]

React розроблений Facebook і постійно оновлюється та підтримується

спільнотою розробників. Однією з головних переваг React є використання Virtual DOM (віртуального DOM), що дозволяє розробникам писати код на JavaScript і повністю контролювати взаємодію з елементами сторінки. Він забезпечує швидке відображення змін та мінімізує кількість звернень до реального DOM, що дозволяє значно покращити продуктивність веб-додатків.

React дозволяє розробникам створювати ієрархію компонентів, що дозволяє значно полегшити процес розробки та забезпечує більш зрозумілий код. Кожен компонент може бути перевикористаний в будь-якому місці додатку, що спрощує розробку та зменшує кількість дублювання коду. (рисунок 2.1)

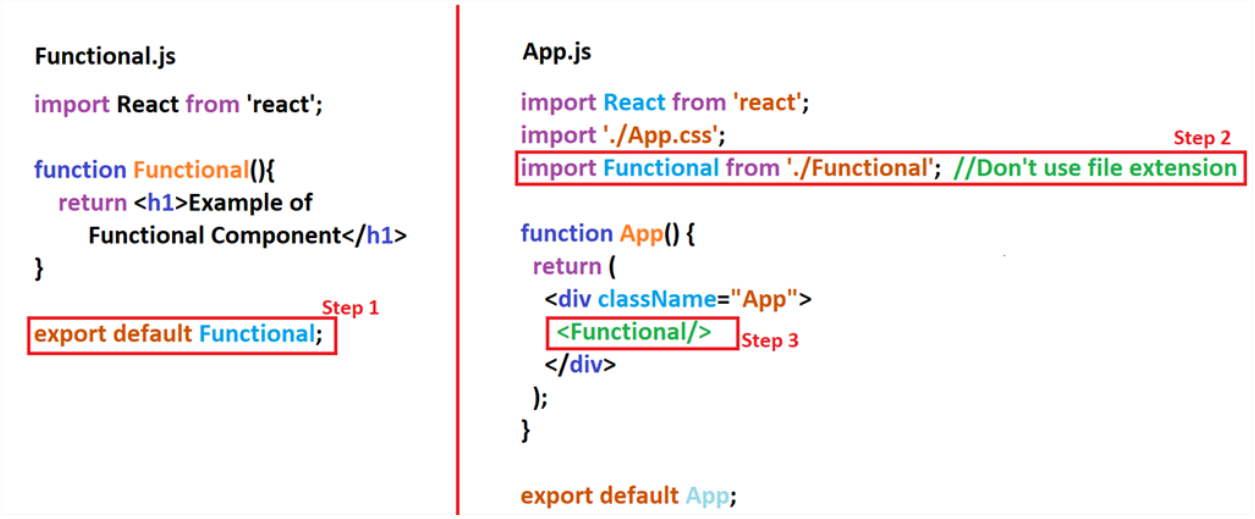

Рисунок. 2.1 – Приклад використання компоненту у React

React використовує JSX – розширення синтаксису JavaScript, яке дозволяє вбудовувати HTML-подібні теги безпосередньо в код JavaScript. Це дозволяє створювати більш зрозумілий код та забезпечує більш просту інтеграцію з HTML.

Крім того, React має велику кількість додаткових бібліотек і плагінів, що допомагають збільшити продуктивність, зробити код більш зрозумілим та зменшити час розробки. Наприклад, Redux дозволяє керувати станом стану додатку, а React Router допомагає з управлінням маршрутизацією та

навігацією в додатку.

Для того, щоб почати працювати з React, розробник повинен мати базові знання JavaScript, HTML та CSS. React можна використовувати з різними редакторами коду, такими як Visual Studio Code, Sublime Text або Atom. Для розробки React-додатків можна використовувати багато інструментів, таких як Create React App, Next.js, Gatsby і багато інших.

Однією з головних переваг React є його підтримка великою спільнотою розробників. Це означає, що в разі виникнення проблеми з React, завжди можна знайти відповідну допомогу в мережі. Крім того, багато статей, документації, туторіалів та відеоуроків доступні безкоштовно в Інтернеті, що дозволяє вивчити React на рівні, який необхідний для користувача.

Узагалі, React є потужною бібліотекою для створення інтерфейсів користувача на веб-сайтах та веб-додатках. Він дозволяє розробникам швидко створювати високопродуктивні та динамічні додатки, забезпечуючи користувачам більш якісний досвід взаємодії з сайтом. Завдяки його ієрархії компонентів, використання Virtual DOM та JSX, розробка додатків на React стає більш простою та зрозумілою. Крім того, велика кількість додаткових бібліотек та плагінів дозволяє розробникам збільшувати продуктивність, зробити код більш зрозумілим та зменшити час розробки.

2.3. Firebase

Firebase – це платформа, розроблена компанією Google, яка надає серію інструментів для розробки мобільних та веб-додатків. Firebase пропонує різноманітні інструменти для зберігання даних, аутентифікації користувачів, розсилки повідомлень та аналізу статистики взаємодії користувачів з додатком. Firebase є повністю керованою платформою, що означає, що ви можете сконцентруватись на розробці вашого додатку, а Firebase дбає про інфраструктуру та безпеку. [7]

2.3.1. Firebase Firestore Database

Firebase Firestore Database – це сервіс баз даних, який надається Firebase. Firestore – це документальна база даних, яка дозволяє зберігати та організовувати дані у вигляді документів, колекцій та підколекцій. Firestore можна використовувати як з мобільними, так і з веб-додатками.

Firestore дозволяє розробникам зберігати та організовувати дані у структурованому форматі, забезпечуючи легкий доступ до даних та підтримуючи повідомлення в режимі реального часу. Firestore підтримує такі типи даних, як рядки, числа, булеві значення, дати та часи, а також структури даних, такі як масиви та об'єкти. Firestore також підтримує транзакції та запити на вибірку даних. (рисунок 2.2)

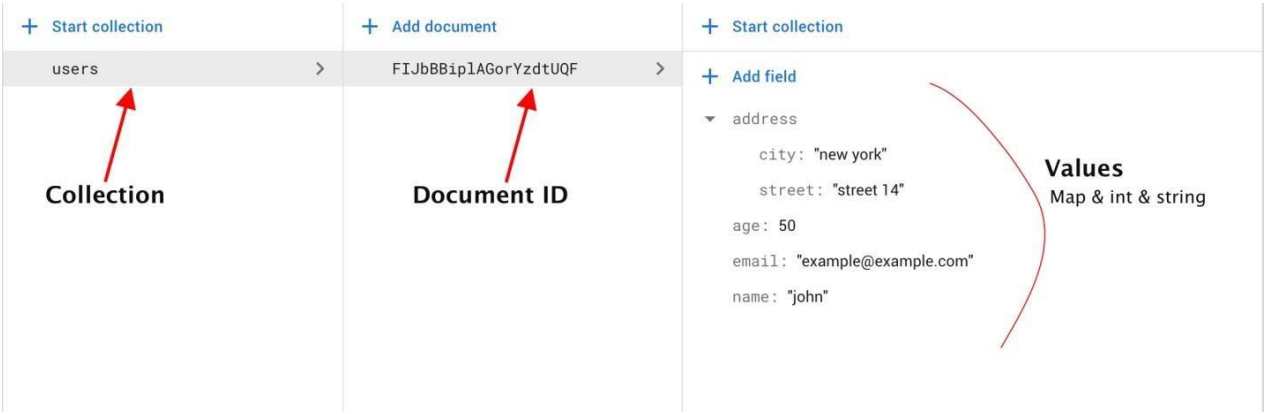

Рисунок 2.2 – Приклад збережених данних у Firestore

Firestore дозволяє легко налаштувати доступ до даних для користувачів, включаючи публічний доступ та доступ за авторизацією. Firestore надає розробникам інструменти для керування правами доступу та захисту даних, такі як правила безпеки. Firestore також підтримує реплікацію даних, що дозволяє забезпечити високу доступність та швидкість роботи бази даних. Реплікація даних означає, що дані зберігаються на декількох серверах у різних регіонах, що дозволяє забезпечити стійкість до відмов серверів та забезпечити швидкий доступ до даних.

Крім того, Firestore дозволяє використовувати різні інструменти для

забезпечення безпеки та конфіденційності даних, такі як правила безпеки (security rules), налаштування доступу та інші. Це дозволяє розробникам забезпечувати високий рівень захисту даних користувачів та дотримуватися вимог законодавства про захист персональних даних.

2.3.2. Firebase Authentication

Firebase Authentication – це сервіс аутентифікації користувачів в додатках, який дозволяє розробникам забезпечувати безпеку та захист даних користувачів. Firebase Authentication підтримує аутентифікацію через різні соціальні мережі, такі як Facebook, Google, Twitter, GitHub та інші, а також дозволяє реалізувати власну систему аутентифікації.

Для роботи з Firebase Authentication не потрібно розуміти складні алгоритми шифрування та безпеки, сервіс надає готові інструменти для захисту даних користувачів. Firebase Authentication також підтримує двофакторну аутентифікацію та інші методи захисту.

Firebase Authentication дозволяє розробникам легко інтегрувати сервіс аутентифікації в свої додатки. Наприклад, можна дозволити користувачам увійти в додаток за допомогою свого облікового запису Google або Facebook, що дозволяє спростити процес реєстрації та входу для користувачів, а також зберігає їхні дані в безпечному місці. Firebase Authentication також дозволяє налаштовувати права доступу користувачів до певних ресурсів додатку.

Крім того, Firebase Authentication надає можливість автоматично відновлювати паролі користувачів та надсилати їм повідомлення електронної пошти для підтвердження електронної адреси. Це допомагає забезпечити безпеку та підтримувати актуальність даних користувачів.

Firebase Authentication має документацію та підтримку, яка дозволяє розробникам швидко зрозуміти, як працювати з сервісом та вирішити будь-які питання, які можуть виникнути під час розробки.

Одним з недоліків Firebase Authentication може бути те, що залежно від обсягу додатку та кількості користувачів, можуть виникнути додаткові витрати на обслуговування та зберігання даних. Проте, Firebase Authentication надає певну кількість безкоштовного обсягу ресурсів, що дозволяє розробникам спробувати та перевірити сервіс перед тим, як зробити вирішальне рішення щодо використання його в своєму додатку.

2.4 Середовище розробки VS Code

VS Code (Visual Studio Code) – це безкоштовне інтегроване середовище розробки, яке надає широкі можливості для (ua5.org) роботи з різними мовами програмування та технологіями. VS Code розроблений компанією Microsoft і є одним з найбільш популярних середовищ розробки у світі. [5]

Однією з особливостей VS Code є його легкість використання та гнучкість. Він має простий та зрозумілий інтерфейс, який дозволяє швидко налаштувати середовище для потреб розробника. Також VS Code має велику кількість плагінів та розширень, що дозволяє налаштувати середовище для роботи з будь-якою мовою програмування та технологією.

Однією з переваг VS Code є його інтеграція з Git. За допомогою вбудованого Git плагіна, розробник може працювати з репозиторіями Git, комітити зміни та пушити їх на віддалений сервер. Крім того, VS Code підтримує роботу з GitHub, що дозволяє розробникам працювати зі своїми проектами на цій платформі безпосередньо з VS Code.

VS Code також має вбудований дебагер, який дозволяє розробникам легко відлагоджувати свій код. Дебагер VS Code підтримує багато мов програмування, включаючи (kanapps.pro) JavaScript, TypeScript, Python, PHP та інші.

Іншою перевагою VS Code є його можливість розширення. Розробники можуть створювати власні плагіни та розширення, що дозволяє налаштувати VS Code для своїх потреб. Крім того, VS Code має велику кількість сторонніх

плагінів та розширень, (kanapps.pro) що дозволяє розширити можливості середовища розробки.

Інтерфейс середовища наведено на рисунку 2.3.

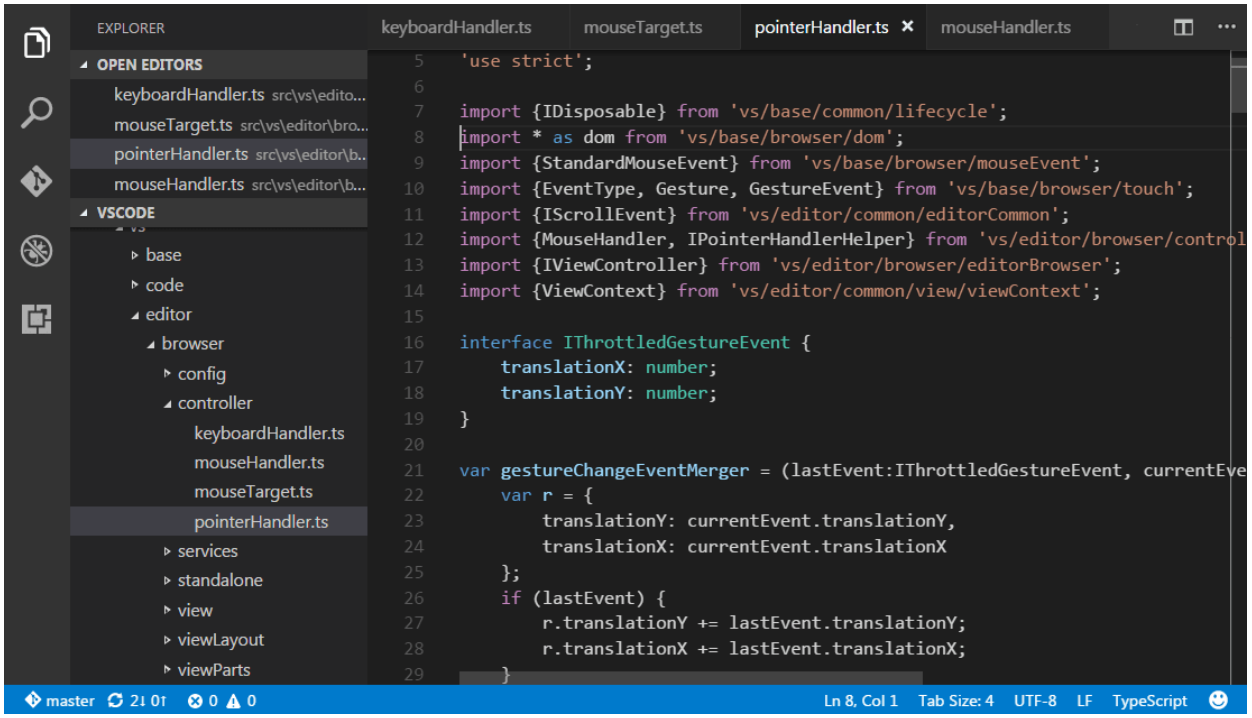

Рисунок 2.3 – Інтерфейс VS Code

Існує багато різноманітних плагінів для VS Code, що дозволяють працювати з різними мовами програмування, фреймворками, бібліотеками та іншими технологіями. Наприклад, плагін Python дозволяє працювати з мовою Python, плагін React Native дозволяє розробляти мобільні додатки для (reposit.nupp.edu.ua) платформи React Native, а плагін Docker дозволяє працювати з контейнерами Docker.

Окрім того, VS Code дозволяє інтегруватись з різними інструментами розробки, такими як системи контролю версій, збірники проектів, тестові фреймворки та інші. Це дозволяє розробникам зручно працювати зі своїми проектами та забезпечує високу продуктивність роботи.

Крім того, VS Code має можливість налаштування різних параметрів та налаштувань, що дозволяє налаштувати робоче середовище під свої потреби.

Наприклад, користувачі можуть змінювати теми оформлення, налаштовувати різні клавіатурні скорочення, налаштовувати автоматичну підсвічування синтаксису та багато іншого.

Отже, VS Code є потужним та гнучким інструментом для розробки програмного забезпечення. Він має велику кількість функцій та можливостей, що дозволяє розробникам налаштовувати його для своїх потреб.

2.5. Висновки за другим розділом

У другому розділі детально розглянуті основні компоненти Wiki-ресурсу, а саме фронтенд- та бекенд-засоби. Аналіз засобів розробки дозволив визначитися з програмним забезпеченням для виконання роботи, зокрема було прийняте рішення використати такі технології, як HTML, SCSS, JavaScript (зокрема ReactJS), та Firebase. В якості середовища розробки оптимальним рішенням буде використання VS Code.

За допомогою вибраних засобів можна створювати потужні веб-додатки, які зможуть забезпечити довільний функціонал. Зокрема, інтерфейс користувача можна розробляти за допомогою ReactJS, а бекенд-підтримку забезпечити за допомогою Firebase. Всі ці компоненти можна легко задіяти за допомогою середовища VS Code, який є потужним та гнучким інструментом для розробки програмного забезпечення.

Загалом, створення веб-ресурсу на базі Wiki-технологій є ефективним та зручним варіантом для створення тематичного веб-ресурсу, такого як тематичний ресурс про всесвіт “Відьмака”. Використання таких технологій, як ReactJS та Firebase, дозволяє розробити дуже потужний та функціональний веб-додаток, який зможе задовольнити потреби користувачів.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ WIKI-ЕНЦИКЛОПЕДІЇ "ВСЕСВІТ ВІДЬМАКА"

3.1. Тематика. Всесвіт "Відьмака"

Сага про відьмака – один із найвідоміших циклів у фентезійній літературі. Це постмодерністське фентезі, створене польським письменником Анджеєм Сапковським, де переосмислюються традиційні сюжети, а характери персонажів не відповідають прийнятим канонам. "Відьмак" – це класична історія про порятунок світу, що порушує важливі етичні та філософські питання.

Всесвіт "Відьмака" є унікальним і складається з різноманітних елементів, таких як магія, монстри, політичні конфлікти та складні персонажі.

Тематика "Всесвіту Відьмака" включає в себе наступні складові:

1) Історія та міфологія. Всесвіт "Відьмака" має глибоко розроблену історію та міфологію. Ця тематика охоплює походження всесвіту, легенди, міфи та історичні події, які вплинули на його сучасний стан.
2) Персонажі. Всесвіт "Відьмака" населений різноманітними персонажами – від відьмаків, які полюють на монстрів, до волхвів, магів, королів та інших. Опис персонажів включає їхню біографію, характеристики, мотивації та взаємини з іншими героями.
3) Місцевість та локації. Всесвіт "Відьмака" розташований у фантастичному середньовічному світі з різноманітними локаціями, такими як міста, села, ліси, гори та замки. Опис місцевості та локацій дозволяє краще зрозуміти атмосферу та культуру всесвіту.
4) Культура та соціальні аспекти. Всесвіт "Відьмака" має свою унікальну культуру, соціальну структуру та норми поведінки. Ця тематика включає опис традицій, релігійних переконань, етики та

інших соціальних аспектів.

5) Магія та монстри. У всесвіті "Відьмака" магія грає важливу роль, а монстри населяють його безліч. Опис магії включає систему магічних здібностей, ритуали, заклинання та магічні об'єкти. Також, опис монстрів дозволяє краще зрозуміти їхні характеристики, поведінку та способи боротьби з ними.

Ці аспекти тематики дозволять створити детальну та оригінальну Wiki-енциклопедію, яка зможе задовольнити інтереси шанувальників всесвіту "Відьмака" та допоможе краще розуміти його комплексність та глибину.

3.2. Призначення ресурсу

Враховуючи насиченість персонажами та стійкий інтерес до творів циклу, логічним є створення зручного довідково-інформаційного ресурсу, що дозволяє користувачам більше дізнатися про ігровий всесвіт і зручно відшукувати потрібну інформацію. Для досягнення цієї мети була розроблена комфортна навігація з глобальним пошуком, яка дозволяє швидко переглядати всі статті на ресурсі (рисунок 3.1)

Рисунок 3.1 – Глобальний пошук

Статті розділені на різні категорії, такі як "Персонажі", "Локації", "Бестіарій". Це допомагає користувачам зручно фільтрувати контент і швидко знаходити інформацію, яка їх цікавить. Наприклад, в категорії "Персонажі" та "Локації" можуть бути доступні фільтри, такі як "Фракція", що дозволяє зорієнтуватися в обраній групі персонажів або місць.

Рисунок 3.2 – Навігація по категоріям

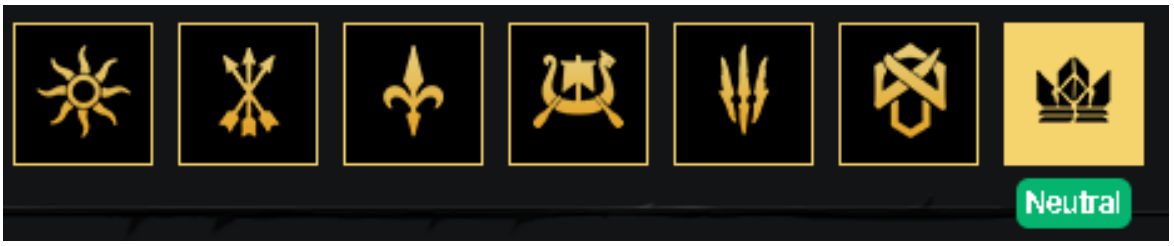

Рисунок 3.3 – Фільтр “Фракція”

Ресурс також надає можливість авторизованим користувачам додавати та редагувати статті (рисунок 3.4) Це створює можливість спільної роботи над наповненням та покращенням контенту. Користувачі можуть створювати свої власні облікові записи та авторизовуватися на веб-ресурсі для здійснення редагувань. У вкладці "Мої статті" користувачі можуть переглянути список статей, які вони додали або відредагували, що допомагає їм відстежувати свої внески (рисунок 3.6)

Рисунок 3.4 – Форма додавання статті аккаунту

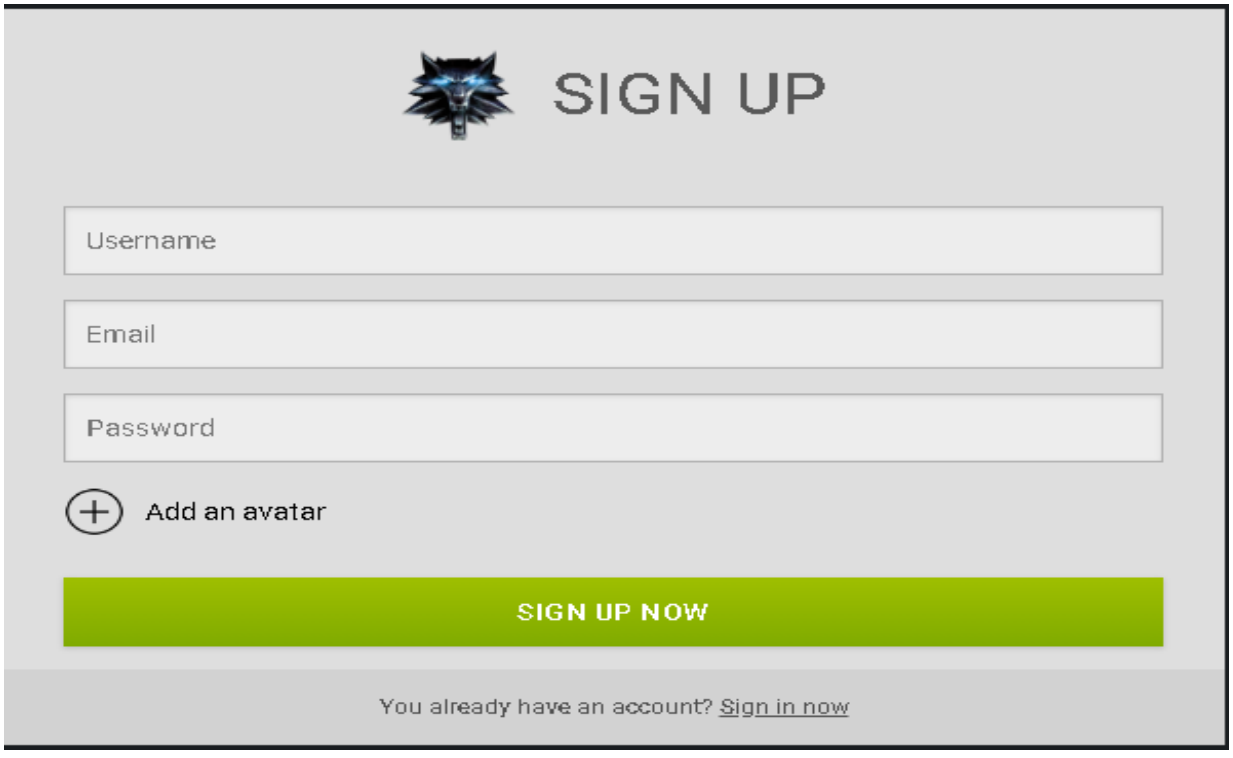

Рисунок 3.5 – Форма створення

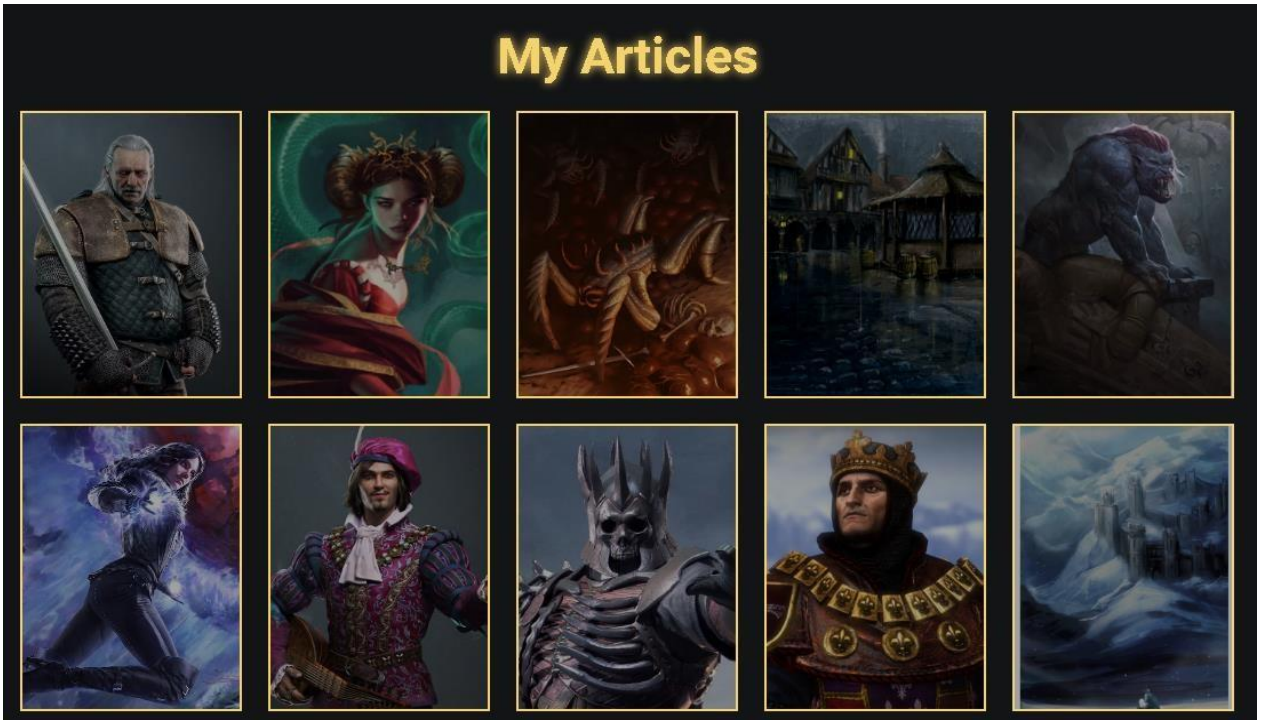

Рисунок 3.6 – Сторінка "Мої статті"

На сторінці кожної статті може бути блок "Contributors", в якому відображаються користувачі, які внесли внесок до цієї статті. Це дає визнання активним учасникам спільноти та надає можливість побачити, хто допоміг зібрати інформацію для конкретної статті (рисунок 3.7)

Рисунок 3.7 – Блок "Contributors"

Загалом, ресурс "Всесвіт Відьмака" створений з метою надання зручного способу отримання інформації про всесвіт "Відьмака". Він забезпечує зручну навігацію, глобальний пошук, можливість спільного редагування, перегляду внесків користувачів, що створює живу та активну спільноту фанатів "Відьмака".

3.3. Структура бази даних

База даних включає наступні колекції (рисунок 3.8):

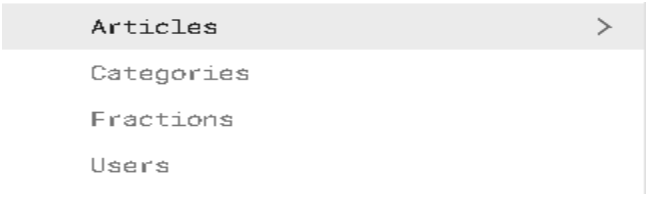

Рисунок 3.8 – Колекції бази даних

Articles (Статті) – це колекція, що містить документи статей з наступними полями (рисунок 3.9):

1) cats (категорії): масив категорій, до яких належить стаття;
2) content (зміст): текстове поле, що містить вміст статті;
3) contributors (учасники): масив користувачів, які внесли внесок до статті;
4) faction (фракція): поле, що містить назву фракції, пов'язаної зі статтею;
5) mainImage (головне зображення): URL головного зображення, пов'язаного зі статтею;
6) title (назва): назва статті.

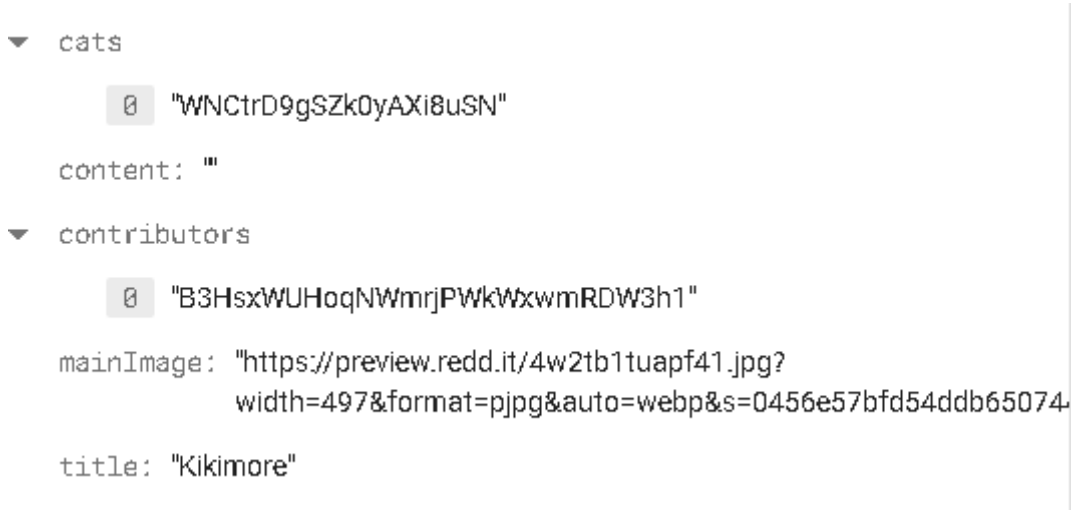

Рисунок 3.9 – Документ із “Articles”

Categories (Категорії) – це колекція, що містить документи категорій з наступними полями (рисунок 3.10):

1) title (назва): назва категорії;
2) articles (статті): підколекція статей з документами, які належать до цієї категорії. Кожен документ містить поле articleRef, яке містить ідентифікатор статті, пов'язаної з категорією.

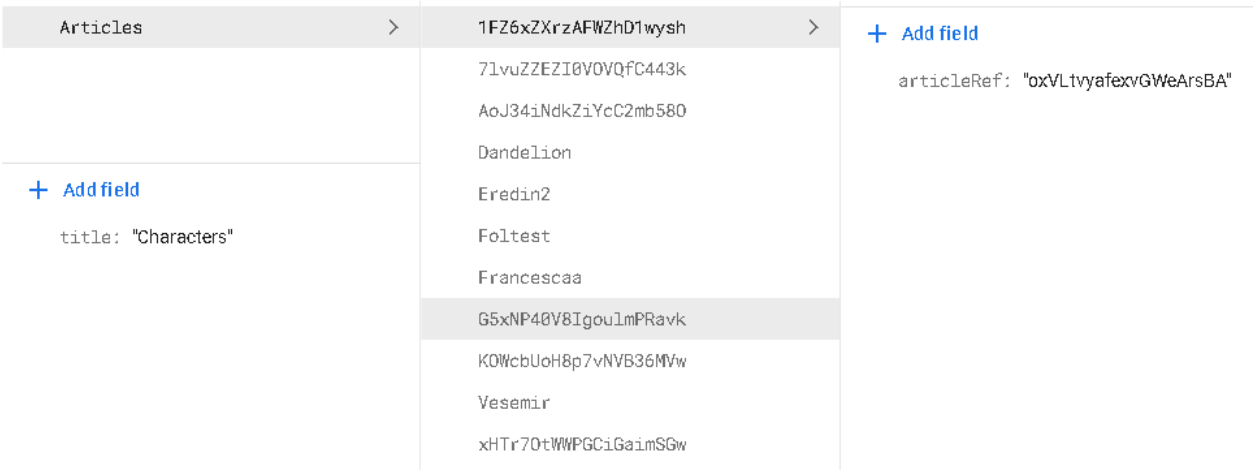

Рисунок 3.10 – Документ із “Categories”

Factions (Фракції) – це колекція, що містить документи фракцій з

наступним полем (рисунок 3.11):

title (назва): назва фракції

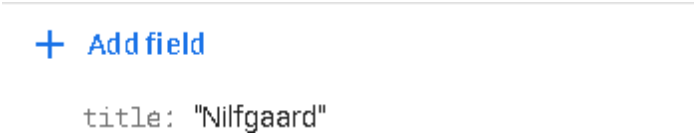

Рисунок 3.11 – Документ із “Factions”

Users (Користувачі) – це колекція, що містить документи користувачів з наступними полями (рисунок 3.12):

1) articles (статті): масив статей, створених або відредагованих користувачем;
2) email (електронна пошта): електронна адреса користувача;
3) img (зображення): URL зображення користувача;
4) password (пароль): хеш пароля користувача;
5) username (ім'я користувача): ім'я користувача.

Рисунок 3.12 – Документ із “Users”

Це загальна структура бази даних, яка дозволить зберігати та організовувати статті, категорії, фракції та користувачів Wiki-енциклопедії "Всесвіт Відьмака". Можна розширювати її функціональність та додавати необхідні поля чи взаємозв'язки, враховуючи специфіку проекту, що створюється.

3.4. Основні програмні модулі

У цьому розділі зосередимося на програмних модулях, які використовуються для відображення різних сторінок та компонентів проекту. Кожен модуль представляє окрему сторінку зі своїм функціоналом та компонентами, які використовуються для відображення вмісту та взаємодії з користувачем. Додатково, розглянемо важливі контексти та кастомні хуки, які забезпечують обмін даними та функціональність між компонентами.

Основні програмні модулі (рисунок 3.13):

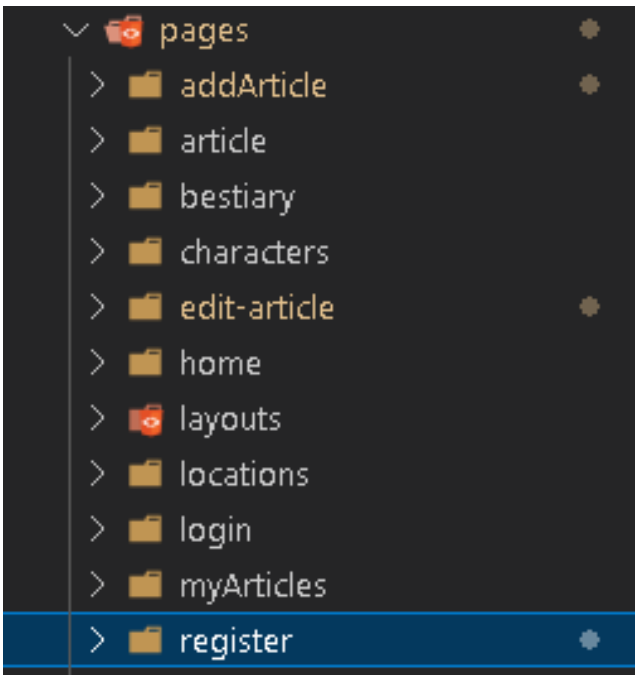

Рисунок 3.13 – Основні програмні модулі

1) AddArticle: Цей модуль відповідає за сторінку додавання нової статті. Він включає компоненти, такі як Input (поле вводу), MulSelect (мульти-вибір), FactionsFilter (фільтр фракцій) та Error (помилки), які допомагають користувачам ввести та обрати необхідну інформацію для статті.
2) Article: Цей модуль представляє сторінку окремої статті. Він включає компоненти, такі як ArticleContent (вміст статті), Loading (завантаження) та Error (помилки), які забезпечують відображення статті, індикацію завантаження та обробку помилок.
3) Bestiary: Цей модуль відповідає за сторінку бестиарію, де користувачі можуть переглядати інформацію про різних істот зі світу "Відьмака". Він включає компоненти, такі як Articles (список статей), які відображають список статей з бестиарію.
4) Characters: Цей модуль представляє сторінку персонажів, де користувачі можуть переглядати інформацію про різних персонажів зі світу "Відьмака". Він включає компоненти, такі як FactionsFilter (фільтр фракцій) та Articles (список статей), які допомагають користувачам фільтрувати та відображати статті про персонажів.
5) Locations: Цей модуль представляє сторінку локацій, де користувачі можуть переглядати інформацію про різні місця зі світу "Відьмака". Він включає компоненти, такі як FactionsFilter (фільтр фракцій) та Articles (список статей), які допомагають користувачам фільтрувати та відображати статті про локації.
6) EditArticle: Цей модуль відповідає за сторінку редагування статті. Він включає компоненти, такі як Input (поле вводу), MulSelect (мульти-вибір), FactionsFilter (фільтр фракцій) та Error (помилки), які допомагають користувачам змінювати та оновлювати вміст статті.

7) Home: Цей модуль представляє домашню сторінку проекту. Він включає компоненти, такі як CatSection (розділ категорій), які відображають список категорій та посилання на відповідні сторінки.
8) Login: Цей модуль відповідає за сторінку входу в систему. Він включає компоненти, такі як Form (форма), Input (поле вводу) та Error (помилки), які допомагають користувачам ввійти до системи та обробляти помилки.
9) Register: Цей модуль представляє сторінку реєстрації нового користувача. Він включає компоненти, такі як Form (форма), Input (поле вводу), Error (помилки) та AddFile (додавання файлу), які допомагають користувачам зареєструватись та обробляти помилки.
10) MyArticles: Цей модуль відповідає за сторінку "Мої статті", де користувачі можуть переглядати свої статті. Він включає компоненти, такі як Articles (список статей), Loading (завантаження) та Error (помилки), які забезпечують відображення та управління статтями користувача.

У проекті також присутній загальний компонент Layout з бібліотеки react-router-dom [8] для відображання загальних компонентів на усіх сторінках, що включає Header (верхню панель), Footer (нижню панель) та Outlet (зону відображення сторінок) (рисунок 3.14).

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

import Header from '../../components/header/Header';
import Footer from '../../components/footer/Footer';

const Main = () => {
  return (
    <>
      <Header />
      <Outlet />
      <Footer />
    </>
  );
};

export default Main;
```

Рисунок 3.14 – Компонент Layout

Крім того, є папка context, де знаходяться важливі контексти які забезпечують обмін даними та функціональність між компонентами, такі як ArticlesContext (контекст статей), AuthContext (контекст авторизації), SearchContext (контекст пошуку) та FilterContext (контекст фільтрації) (рисунок 3.15).

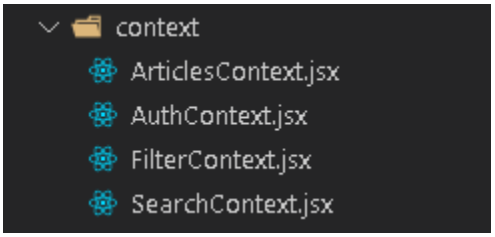

Рисунок 3.15 – Контексти для обміну даних

У папці hooks можна знайти кастомні хуки, які забезпечують додаткову функціональність додатку (рисунок 3.16).

Рисунок 3.16 – Кастомні хуки

Наприклад, useFetchArticlesByCat використовується для отримання статей з певної категорії, useFetchDocsFromColl допомагає отримати документи з певної колекції, а useUploadFile використовується для завантаження файлів.

Ці основні програмні модулі, контексти та кастомні хуки допомагають побудувати функціональний та ефективний веб-ресурс на базі Wiki-технологій зі сторінками, компонентами та функціями, які дозволяють користувачам взаємодіяти зі світом "Відьмака" та знаходити необхідну інформацію.

3.5. Розгортання системи

Для розгортання системи було використано платформу Vercel [9] разом з інтеграцією з GitHub. Цей підхід дозволив швидко та зручно розгорнути веб-ресурс із проекту на хостингу та забезпечити його доступність для користувачів.

Основні кроки розгортання системи на Vercel з використанням GitHub:

1) Підготовка репозиторію: У репозиторії проекту було створено файл vercel.json, в якому була вказана конфігурація для розгортання на Vercel.

2) Інтеграція з Vercel: На платформі Vercel було створено новий проект та підключено репозиторій з кодом проекту з GitHub. Для цього була використана автоматична інтеграція, що дозволяє автоматично розгортати проект після змін у відповідній гілці репозиторію.
3) Налаштування змінних середовища: При розгортанні системи на Vercel, змінні середовища, які зазвичай зберігаються в файлі .env, були вказані безпосередньо на платформі. Це дозволяло передавати необхідну інформацію про Firebase, яка була збережена у файлі .env на комп'ютері, без включення цих змінних до репозиторію на GitHub.
4) Розгортання системи: Після налаштування проекту та змінних середовища розгортання системи на Vercel було запущено. Платформа автоматично підтягувала останню версію коду з GitHub та розгортала систему на основі налаштувань, вказаних у vercel.json.
5) Перевірка та тестування: Після успішного розгортання системи на Vercel проводилося перевірка та тестування веб-ресурсу, щоб переконатися в правильному функціонуванні сторінок та компонентів.

В результаті цього процесу веб-ресурс був успішно розгорнутий на платформі Vercel та став доступним для відвідувачів. За необхідності, можна вносити зміни до проекту на локальному комп'ютері, а потім оновлювати розгорнутий веб-ресурс на Vercel знову, з повторенням вищезазначених кроків.

Цей підхід до розгортання дозволяє зручно та ефективно керувати проектом та забезпечити його актуальність та доступність для користувачів у найкоротші терміни.

ВИСНОВКИ

В даній дипломній роботі був розроблений тематичний веб-ресурс на базі Wiki-технологій, присвячений всесвіту "Відьмака". Робота включала аналіз сучасних рішень, проектування та реалізацію системи, яка надає зручний доступ до інформації про персонажів, локації та інші елементи цього всесвіту.

Основним призначенням розробленого ресурсу є надання користувачам можливості ознайомитися з різноманітними аспектами всесвіту "Відьмака". Для цього була створена зручна навігація з глобальним пошуком та категоріями, які дозволяють знайти необхідну інформацію швидко та ефективно. Крім того, була реалізована можливість авторизації, що дозволяє авторизованим користувачам додавати та редагувати статті.

Ключові сторінки проекту були реалізовані з використанням React та React Router, крім того, в роботі використані контексти та кастомні хуки, які сприяють зручному та ефективному управлінню даними та функціональністю ресурсу.

Для розгортання системи була використана платформа Vercel разом з інтеграцією з GitHub. Цей підхід дозволяє автоматично розгортати проект на хостингу та забезпечувати його доступність для користувачів. Крім того, змінні середовища, що містять конфіденційну інформацію, були вказані безпосередньо на платформі під час розгортання.

В цілому, розроблений тематичний веб-ресурс забезпечує зручну та доступну інформацію про всесвіт "Відьмака" для користувачів. Він демонструє використання сучасних технологій, таких як React, Firebase та Vercel, для створення функціональності та забезпечення надійності та ефективності ресурсу.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MediaWiki [Електронний ресурс] / Режим доступу: www. URL: https://www.mediawiki.org/wiki/Manual:What_is_MediaWiki%3F/ru – 29.03.2023 р.
2. DokuWiki [Електронний ресурс] / Режим доступу: www. URL: https://www.dokuwiki.org/dokuwiki – 30.03.2023 р.
3. WikkaWiki [Електронний ресурс] / Режим доступу: www. URL: http://wikkawiki.org/HomePage – 01.04.2023 р.
4. Tiki Wiki CMS Groupware [Електронний ресурс] / Режим доступу: www. URL: https://tiki.org/HomePage – 01.04.2023 р.
5. VS Code [Електронний ресурс] / Режим доступу: www. URL: https://code.visualstudio.com – 15.04.2023 р.
6. React [Електронний ресурс] / Режим доступу: www. URL: https://ru.legacy.reactjs.org – 19.03.2023 р.
7. Firebase [Електронний ресурс] / Режим доступу: www. URL: https://firebase.google.com – 19.03.2023 р.
8. React Router [Електронний ресурс] / Режим доступу: www. URL: https://reactrouter.com/en/main - 29.04.2023 р.
9. Vercel [Електронний ресурс] / Режим доступу: www. URL: https://vercel.com - 29.04.2023 р.
10. Введение в HTML [Електронний ресурс] / Режим доступу: www. URL: https://developer.mozilla.org/ru/docs/Learn/HTML/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B2_HTML – 29.04.2023 р.
11. Современный учебник JavaScript [Електронний ресурс] / Режим доступу: www. URL: https://learn.javascript.ru/ – 29.04.2023 р.
12. Справочник CSS [Електроний ресурс] / Режим доступу: www. URL: https://webref.ru/css – 29.04.2023 р.

13. JavaScript [Електроний ресурс] / Режим доступу: www. URL: https://developer.mozilla.org/ru/docs/Learn/JavaScript – 29.04.2023 р.
14. Мэтью Д. HTML5. Разработка веб-приложений. [Текст] / Д. Мэтью М.: Рид Групп, 2017 – 320 с.
15. Мейер Э., Уэйл Э. CSS: полный справочник / Э. Мейер, Э. Уэйл. – СанктПетербург: ООО „Диалектика”, 2019. – 1088 с.

Додаток А

Вихідний код

```
import { useEffect, useState } from 'react';
import { db } from '../config/firebase.js';
import { getDocs, collection } from 'firebase/firestore';
import getDocById from '../utils/getDocById';

const useFetchArticlesByCat = (cat) => {
  const [articles, setArticles] = useState([]);
  const [isLoading, setIsLoading] = useState(false);
  const [fetchError, setFetchError] = useState(null);

  const catsRef = collection(db, 'Categories');

  useEffect(() => {
    const getArticlesByCat = async (categories) => {
      try {
        setIsLoading(true);

        const data = await getDocs(catsRef);
        const cats = data.docs.map((doc) => ({
          ...doc.data(),
          id: doc.id,
        }));

        const chosenCat = cats.find((cat) => cat.title === categories);
        const catArticlesRef = collection(
          db,
          'Categories',
```

```
        chosenCat.id,
        'Articles'
      );

      const articlesDocs = await getDocs(catArticlesRef);
      const articlesRefs = articlesDocs.docs.map((doc) => ({
        ...doc.data(),
      }));
      const catArticles = await Promise.all(
        articlesRefs.map((articlesRef) =>
        getDocById('Articles', articlesRef.articleRef)
        )
      );

      setArticles(catArticles);
      setFetchError(null);
    } catch (err) {
      setFetchError(err.message);
    } finally {
      setIsLoading(false);
    }
  };

  getArticlesByCat(cat);
 }, [cat]);

 return { articles, fetchError, isLoading };
};
```

Хук useFetchArticlesByCat

```
import { useEffect, useState } from 'react';
import { db } from '../config/firebase.js';
import { getDocs, collection } from 'firebase/firestore';

const useFetchDocsFromColl = (collectionName) => {
  const [data, setData] = useState([]);
  const [isLoading, setIsLoading] = useState(false);
  const [fetchError, setFetchError] = useState(null);

  const collRef = collection(db, collectionName);

  useEffect(() => {
    const getData = async () => {
      try {
        setIsLoading(true);

        const data = await getDocs(collRef);
        const filteredData = data.docs.map((doc) => ({
          ...doc.data(),
          id: doc.id,
        }));

        setData(filteredData);
        setFetchError(null);
      } catch (err) {
        setFetchError(err.message);
      } finally {
        setIsLoading(false);
```

```
    }
   };

   getData();
  }, []);

  return { data, fetchError, isLoading };
};
```

Хук useFetchDocsByColl

```
import { useEffect, useState } from 'react';
import { ref, uploadBytesResumable, getDownloadURL } from 'firebase/storage';
import { storage } from '../config/firebase';

const useUploadFile = (file, path = '') => {
  const [isLoading, setIsLoading] = useState(false);
  const [uploadError, setUploadError] = useState('');
  const [downloadUrl, setDownloadUrl] = useState('');

  useEffect(() => {
   const uploadFile = () => {
    const name = new Date().getTime() + file.name;
    const storageRef = ref(storage, path + name);
    const uploadTask = uploadBytesResumable(storageRef, file);

    uploadTask.on(
     'state_changed',
     () => {
      setIsLoading(true);
```

```
      },
      (error) => {
        setUploadError(error.message);
      },
      async () => {
        const downloadURL = await getDownloadURL(uploadTask.snapshot.ref);
        setDownloadUrl(downloadURL);

        setIsLoading(false);
      }
    );
  };

  file && uploadFile();
 }, [file, path]);

 return { isLoading, uploadError, downloadUrl };
};
```

Хук useUploadFile

```
import React, { createContext, useEffect, useState } from 'react';
import useFetchDocsFromColl from '../hooks/useFetchDocsFromColl';

const ArticlesContext = createContext();

export const ArticlesContextProvider = ({ children }) => {
 const { data, isLoading, fetchError } = useFetchDocsFromColl('Articles');
 const [articles, setArticles] = useState([]);
```

```
  useEffect(() => {
    setArticles(data);
  }, [data]);

  return (
    <ArticlesContext.Provider
      value={{
        articles,
        setArticles,
        isLoading,
        fetchError,
      }}
    >
      {children}
    </ArticlesContext.Provider>
  );
};
```

Контекст ArticlesContext

```
import React, { createContext, useReducer, useEffect } from 'react';
import AuthReducer from '../reducers/AuthReducer';

const INITIAL_STATE = {
  currentUser: JSON.parse(localStorage.getItem('user')) || null,
};

const AuthContext = createContext(INITIAL_STATE);

export const AuthContextProvider = ({ children }) => {
```

```
  const [state, dispatch] = useReducer(AuthReducer, INITIAL_STATE);

  useEffect(() => {
    localStorage.setItem('user', JSON.stringify(state.currentUser));
  }, [state.currentUser]);

  return (
    <AuthContext.Provider value={{ currentUser: state.currentUser, dispatch }}>
      {children}
    </AuthContext.Provider>
  );
};
```