## ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ Навчально-науковий інститут денної освіти Форма навчання денна Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту Завідувач кафедри (підпис)

« » 202 p.

## КВАЛІФІКАЦІЙНА РОБОТА

на тему «РОЗРОБКА ПЛАТФОРМИ ДЛЯ ВЕДЕННЯ БЛОГУ ВИКЛАДАЧА»

зі спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня магістра

Виконавець роботи Комар Ілля Ігорович \_\_\_\_\_ «\_\_\_\_»\_\_\_\_ 202\_ p.

Науковий керівник к.пед.н., доцент, Кошова Оксана Петрівна <u>\_\_\_\_\_ «\_\_\_»</u>\_\_\_\_ 202\_р.

ПОЛТАВА 2025 р.

			ЗАТВЕРДЖ	УЮ
Завідувач кафедри _		Олена	а ОЛЬХОВСЬ	КА
	(п	ідпис)		
	<u> </u>	>>>	202_	_p.

## ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему «Розробка платформи для ведення блогу викладача»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня магістр

Прізвище, ім'я, по батькові Комар Ілля Ігорович

Затверджена наказом ректора № \_\_\_-Н від «\_\_»\_\_\_\_ 202\_ р.

Термін подання студентом роботи «\_\_\_\_»\_\_\_\_ 202\_\_ р.

Вихідні дані до кваліфікаційної роботи: <u>публікації по темі роботи, статті та</u> документації з платформ для ведення блогу викладача.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

- 1. ПОСТАНОВКА ЗАДАЧІ
- 2. ОГЛЯД ПЛАТФОРМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ
  - 2.1. Платформа для ведення блогу Blogger.
  - 2.2. Платформа для ведення блогу Medium.
  - 2.3. Платформа для ведення блогу Ghost.
  - 2.4. Платформа для ведення блогу Tumblr.
- 3. ТЕОРЕТИЧНА ЧАСТИНА
  - 3.1. Опис проектних рішень та інструментів розробки.
  - 3.2. Опис обраного підходу до створення програмного забезпечення.
  - 3.3. Прототип інтерфейсу платформи.
- 4. ПРАКТИЧНА ЧАСТИНА
  - 4.1. Архітектура програмного забезпечення.
  - 4.2. Опис діаграми роботи платформи та розгортання.
  - 4.3. Інструкція з використання платформи.

ВИСНОВКИ

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

ДОДАТОК А

Перелік графічного матеріалу: 78 ілюстрацій.

Розділ	Прізвище,	Підпі	ис, дата
	ініціали, посада	завдання	завдання
	консультанта	видав	прийняв
Постановка задачі	Кошова О.П.		
Інформаційний огляд	Кошова О.П.		
Теоретична частина	Кошова О.П.		
Практична частина	Кошова О.П.		

# Консультанти розділів кваліфікаційної роботи

Календарний графік виконання кваліфікаційної роботи

	Зміст роботи	Термін	Фактичне
		виконання	виконання
1.	Вступ		
2.	Вивчення методичних рекомендацій та		
ста	андартів та звіт керівнику		
3.	Постановка задачі		
4.	Інформаційний огляд джерел бібліотек та		
інт	гернету		
5.	Теоретична частина		
6.	Практична частина		
7.	Закінчення оформлення		
8.	Доповідь студента на кафедрі		
9.	Доробка (за необхідністю), рецензування		

Дата видачі завдання «	» 202_	p.			
Здобувач вищої освіти		Комар Ілля Ігорович			
	(підпис)				
Науковий керівник		к.пед.н., доц. Кошова О.П.			
	(підпис)	(науковий ступінь, вчене звання, ініціали та прізвище)			
<b>Результати захисту кваліфікаційної роботи</b> Кваліфікаційна робота оцінена на					

		(балів, оці	нка за національною шкал	лою, оцінка за ЕСТЅ)
Протокол засідання ЕК № _		від «	>>	202_ p.
Секретар ЕК				
	(підпис)	_	(ініціали та прізвище)	

## Затверджую Зав. кафедрою \_\_\_\_\_ к.ф.-м.н. Олена ОЛЬХОВСЬКА «\_\_\_\_» \_\_\_\_ 202\_\_ р.

**Погоджено** Науковий керівник \_\_\_\_\_ к.пед.н., Оксана КОШОВА «\_\_\_\_» \_\_\_\_\_ 202\_\_\_ р.

## План

кваліфікаційної роботи на тему «Розробка платформи для ведення блогу викладача» зі спеціальності 122 Комп'ютерні науки освітня програма 122 «Комп'ютерні науки» ступеня магістр Прізвище, ім'я, по батькові <u>Комар Ілля Ігорович</u>

# ВСТУП

1. ПОСТАНОВКА ЗАДАЧІ

# 2. ОГЛЯД ПЛАТФОРМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ

- 2.1. Платформа для ведення блогу Blogger.
- 2.2. Платформа для ведення блогу Medium.
- 2.3. Платформа для ведення блогу Ghost.
- 2.4. Платформа для ведення блогу Tumblr.

# 3. ТЕОРЕТИЧНА ЧАСТИНА

- 3.1. Опис проектних рішень та інструментів розробки.
- 3.2. Опис обраного підходу до створення програмного забезпечення.
- 3.3. Прототип інтерфейсу платформи.
- 4. ПРАКТИЧНА ЧАСТИНА
  - 4.1. Архітектура програмного забезпечення.
  - 4.2. Опис діаграми роботи платформи та розгортання.
  - 4.3. Інструкція з використання платформи.

# ВИСНОВКИ

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

# ДОДАТОК А

Здобув	ач вищої освіти	Ілля КОМАР
«»	202p.	

#### РЕФЕРАТ

Записка: 101 сторінка, 78 рисунків, 1 додаток, 22 літературних джерел.

Мета роботи – розробка платформи для ведення блогу викладача.

Об'єкт розробки – платформа для ведення блогу викладача.

Методи дослідження – використання технологій та інструментів розробки програмного забезпечення: сервіс для розробки інтерфейсів Figma, фреймворк Next.js, мова програмування TypeScript, база даних MongoDB, бібліотека React, редактор коду Visual Studio Code, розподілена система контролю версій Git.

Сформульовано вимоги до програмного забезпечення. Зроблено огляд платформ аналогічного призначення, виділені їх основні переваги та недоліки. Виконано опис проектних рішень та інструментів до розробки програмного забезпечення. Обрано методологію створення програмного забезпечення. інтерфейсу. Розроблено прототип Описано архітектуру програмного забезпечення. Побудовано діаграму прецедентів. Описано процедуру розгортання. Розроблено блог-платформу та інструкцію.

Ключові слова: ПЛАТФОРМА, БЛОГ, ВИКЛАДАЧ, ОСВІТНІЙ ПРОЦЕС.

# **3MICT**

ВСТУП	7
1. ПОСТАНОВКА ЗАДАЧІ	9
2. ОГЛЯД ПЛАТФОРМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ	
2.1. Платформа для ведення блогу Blogger	
2.2. Платформа для ведення блогу Medium	16
2.3. Платформа для ведення блогу Ghost	
2.4. Платформа для ведення блогу Tumblr	
3. ТЕОРЕТИЧНА ЧАСТИНА	
3.1. Опис проектних рішень та інструментів розробки	
3.2. Опис обраного підходу до створення програмного забезпечення.	
3.3. Прототип інтерфейсу платформи	
4. ПРАКТИЧНА ЧАСТИНА	
4.1. Архітектура програмного забезпечення	
4.2. Опис діаграми роботи платформи та розгортання	64
4.3. Інструкція з використання платформи	71
ВИСНОВКИ	
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	
ДОДАТОК А. ВИХІДНІ КОДИ	

#### вступ

Актуальність. Розробка блог-платформи для освітнього процесу залишається актуальною і сьогодні. В останні роки блоги стали популярним засобом вираження ідей, обміну знаннями та спілкування з іншими людьми. Ведення блогу часто асоціюється з особистими історіями чи розвагами, але для викладачів це також чудовий спосіб ділитися ідеями та працювати разом. Ведення блогу дозволяє викладачам ділитися своїми знаннями, досвідом та практикою з іншими викладачами, студентами та їх батьками. Розробка платформи для ведення блогу дає численні переваги як для самих викладачів, так і для ширшої освітньої спільноти.

Перш за все, блог-платформа для викладачів надає можливість їх професійному розвитку. Викладачі можуть користуватися цією платформою, щоб документувати свій досвід, ділитися планами лекцій, обговорювати практики та розмірковувати над власними стратегіями викладання. Викладачі можуть поглибити своє розуміння різних педагогічних підходів, проаналізувати власні методи викладання та дослідити інноваційні методики.

Крім того, блог-платформа освітнього процесу дозволяє викладачам ділитися ресурсами та ідеями. Викладачі часто присвячують незліченну кількість годин для пошуку ефективних тактик викладання. Платформа дозволяє викладачам без зайвих зусиль ділитися інструментами, які вони знайшли або створили. Платформа дозволяє їм вчитися на досвіді один одного та розвивати ідеї один одного. Такі професійні зв'язки особливо корисні для викладачів, які працюють у віддалених чи ізольованих місцевостях і не мають доступу до можливостей особистої співпраці.

Блог-платформа освітнього процесу дозволяє документувати діяльність та ділитися досягненнями студентів. Викладачі також можуть залучати батьків до навчального процесу та сприяти зміцненню зв'язку між університетом та домом. Крім того, студенти можуть отримати користь від блогу, маючи доступ до додаткових ресурсів, навчальних посібників при виконанні домашніх завдань.

Таким чином, розробка блог-платформи, є надзвичайно актуальною і має багато потенційних переваг. Від надання простору для професійного розвитку до сприяння обміну ресурсами та розбудови освітніх спільнот.

Метою роботи є розробка платформи для ведення блогу викладача.

Об'єктом розробки є платформа для ведення блогу викладача.

Предметом розробки є програмне забезпечення, а саме вебсайти для перегляду та керування блогом.

Кваліфікаційна робота складається з чотирьох розділів, а саме: постановка задачі, огляд платформ аналогічного призначення, теоретичної та практичної частини. Структура побудована так, що дає змогу логічного представлення матеріалу та розкриття теми роботи.

Результатом виконання кваліфікаційної роботи є створення платформи для ведення блогу викладача за допомогою фреймворка Next.js.

Обсяг пояснювальної записки: 101 сторінка, 78 рисунків, 1 додаток, 22 літературних джерела.

#### 1. ПОСТАНОВКА ЗАДАЧІ

Необхідно розробити програмне забезпечення у вигляді вебдодатку та вебсайту для ведення блогу викладача. Перший вебдодаток буде використовуватися для керування контентом блогу через адміністративний інтерфейс, другий – публічний сайт блогу, на якому відвідувачі можуть переглядати публікації викладача. Система повинна забезпечити зручний процес управління публікаціями та їх перегляд у відповідності до сучасних стандартів веброзробки.

Розроблена платформа для ведення блогу викладача повинна складатися з наступних сторінок:

- авторизація;

– статті;

– книги;

- статистика;

- налаштування;

- мій блог.

*Сторінка «Авторизація»*. Процедура авторизації передбачає можливість входу користувача в систему. На сторінці повинна бути розміщена форма, для валідації введених даних і у разі їх підтвердження, надавати доступ до системи.

Сторінка «Статті». Має бути реалізована можливість перегляду всіх статей та керування ними. Для доступу до інформації користувач зможе натискати на потрібну статтю, після чого він потрапить на сторінку з деталями статті.

На сторінці для керування статтями повинні бути доступні дії: перегляд, оновлення та видалення статті.

Для додавання нової статті повинна бути відповідна кнопка, після натискання якої буде відкриття нової сторінки, де буде можливість створення статті та додавання необхідних налаштувань. Сторінка «Книги». Повинні бути реалізовані сторінки для перегляду книг та керування книгами. Для перегляду інформації повинна бути можливість натискання на потрібну книгу, в результаті чого користувач потрапляє на сторінку, де може переглянути та завантажити книгу. Для додавання нової книги повинна бути відповідна кнопка.

Сторінка «Статистика». Має бути реалізована сторінка для перегляду статистики. На сторінці має бути можливість переглядати кількість читачів за весь час, кількість статей та книг. Також повинен бути графік, що демонструє кількість статей за останні 6 місяців.

Сторінка «Налаштування». Має бути реалізована сторінка для редагування профілю, де користувач зможе змінювати особисту інформацію, додавати посилання на соціальні мережі та додаткову інформацію про себе.

Сторінка «Мій блог». Має бути реалізовано можливість переглядати статті, книги та інформацію про автора блогу. Для зміни теми передбачена відповідна кнопка.

## 2. ОГЛЯД ПЛАТФОРМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ

На даний момент у світі існує значне число платформ для ведення блогів, тому далі ми розглянемо деякі з них.

## 2.1. Платформа для ведення блогу Blogger

Blogger – це безкоштовна платформа для ведення блогів, яка дозволяє користувачам легко створювати та керувати власним онлайн-блогом. Була розроблена компанією Руга Labs у 1999 році, а потім придбана компанією Google у 2003 році. Вона є зручною платформою, де користувачі можуть ділитися своїми думками, досвідом та знаннями з усього світу (Рисунок 2.1).



Рисунок 2.1 – Платформа для ведення блогів «Blogger»

Blogger підтримує як однокористувачеві, так і багатокористувачеві блоги та надає багато функцій, такі як: планування публікацій, додавання міток та тегів, розміщення зображень та відео. Blogger став популярним вибором як для початківців, так і для досвідчених блогерів завдяки потужному набору інструментів і простому інтерфейсу [1]. Blogger – це зручна та проста у використанні платформа для ведення блогів. Щоб почати користуватися платформою, необхідно увійти або зареєструватися, використовуючи обліковий запис Google (Рисунок 2.2).

G	
Увійдіть Перейти в Blogger	Електронна адреса або телефон Забули електронну адресу?
	Інший комп'ютер? Щоб увійти в обліковий запис, використовуйте режим гостя. Докладніше про режим гостя
	Створити обліковий запис Далі

Рисунок 2.2 – Вхід в «Blogger» використовуючи обліковий запис Google

Після успішного входу, користувач має вказати назву свого блогу. Назвою блогу може бути ім'я користувача, назва бренду чи будь-яка фраза, яка буде відображати тематику блогу. Далі Blogger надає доменне ім'я, яке закінчується на «blogspot.com». Це URL-адреса за замовчуванням, але є можливість використовувати власне доменне ім'я або доменне ім'я, придбане в Google Доменах.

У платформі Blogger теми відіграють ключову роль у візуальному оформленні блогу, дозволяючи адаптувати його стиль відповідно до уподобань користувача або специфіки обраної тематики. За замовчуванням для новоствореного блогу автоматично встановлюється стандартна тема під назвою Contempo Light. Це мінімалістична, сучасна тема, яка забезпечує чистий і професійний вигляд блогу без необхідності додаткових налаштувань. Однак користувачам доступний широкий вибір інших тем, які пропонують різні дизайни, кольорові схеми та компонування (Рисунок 2.3).

≡	Θ			3		0
Wise	НОВА ПУБЛІКАЦІЯ		Моя тема Толина и толика Толина и толика Сопtempo Light			
	Публікації	l				
ii:	Статистика		e construction de la construcción de la constru La construcción de la const			
	Коментарі	ł				
\$	Прибутки					
	Сторінки	ł				
=	Макет		Contempo			
P	Тема		Here and There	There	ð	
\$	Налаштування		A constant of the second secon			
	Список читання	*				

Рисунок 2.3 – Вибір готових тем на платформі «Blogger»

Також, Blogger надає інструменти для налаштування обраної теми. Користувачі можуть змінювати кольори, шрифти, зображення фону та інші візуальні елементи, що дає можливість зробити блог унікальним (Рисунок 2.4).

< 🕒		0 🗰
Колір фону основного тексту Прозорий фон Кольори тем Ф Ф Ф Ф Ф	<b>Мій блог</b> Все про мене	Q
ДОДАТИ	ГОЛОВНА СТОРІНКА ПРО ПРОЕКТ	
Шрифт За умовчанням В <i>I</i> 16рх	Як обрати комп'ютер	
Колір 📲 Прозорий фон		

Рисунок 2.4 – Налаштування теми на платформі «Blogger»

Щоб створити перший допис, на головній сторінці Blogger потрібно натиснути кнопку «Нова публікація». Спочатку необхідно заповнити поле «Заголовок», що відображає суть статті, а потім написати основний текст. В редакторі доступні інструменти для форматування тексту, додавання зображень, відео, посилань та інших медіа-матеріалів. Для покращення

пошукової оптимізації можна вказати метатеги та ключові слова. Для опублікування потрібно перевірити допис та натиснути кнопку «Опублікувати», щоб він став доступним для читачів (Рисунок 2.5).



Рисунок 2.5 – Створення допису на платформі «Blogger»

Функція статистики є однією з основних переваг Blogger. Вона дозволяє відслідковувати перегляди блогу в режимі реального часу за допомогою графіків, а також надавати інформацію про географічне походження аудиторії, демонструючи, з яких регіонів надходить трафік. Оскільки Blogger є частиною екосистеми Google, є можливість інтегрувати свій блог з Google Analytics (Рисунок 2.6).



Рисунок 2.6 – Перегляд статистики на платформі «Blogger»

Функція монетизації за допомогою Google AdSense надає можливість отримувати дохід від блогу. Для її використання необхідно мати наповнений контентом блог, досягнення 18-річного віку, а в окремих випадках - забезпечити активність сайту протягом шести місяців. За допомогою Google AdSense є можливість налаштувати рекламу на блогові, що стане джерелом додаткового заробітку [1] (Рисунок 2.7).



Рисунок 2.7 – Налаштування монетизації на платформі «Blogger»

#### Переваги користування:

 Простота використання. Blogger має інтуїтивний і простий у використанні інтерфейс, що дозволяє новачкам швидко зрозуміти, як створювати та керувати блогом.

 Безкоштовність. Blogger є безкоштовною платформою, що дозволяє створити та розмістити свій блог без витрат на хостинг або доменне ім'я.

– Інтеграція з Google. Оскільки Blogger є власністю Google, він неперевершено інтегрується з іншими продуктами Google, такими як Google Analytics, AdSense та Google Search Console.

Адаптивний для мобільних пристроїв. Підтримує вбудований мобільний інтерфейс [2].

– Захищена та безпечна платформа. Оскільки Blogger повністю контролюється Google, користувачі повністю позбавлені напруги щодо серверних ресурсів,

контенту, резервного копіювання тощо, тоді як самостійний хостинг є менш захищеним і безпечним.

 Підтримка та допомога. Якщо виникла проблема чи питання, є можливість звернутися за допомогою до служби підтримки;

– Теми. Blogger пропонує близько 20 професійних безкоштовних тем для встановлення на блог [3].

Недоліки користування:

– Втрата власності. Використовуючи Blogger, користувачі мають обмежену власність і контроль над своїм блогом. Google має право припинити або видалити блог, якщо він порушує їхні умови обслуговування.

- Обмежені функціональні можливості. Неможливо встановити плагіни.

 Жахливе форматування. Багато користувачів вважають, що форматування, особливо візуального контенту, у Blogger є жахливим.

#### 2.2. Платформа для ведення блогу Medium

Medium – платформа для ведення блогів, заснована Еваном Вільямсом в серпні 2012 року. Початкова мета полягала в тому, щоб заохотити користувачів створювати довші пости, ніж у Twitter, ліміт яких був всього 280 символів.

Medium є величезною спільнотою читачів і письменників, які об'єднуються, щоб поділитися унікальними поглядами на різні теми від технологічних трендів і бізнес-інсайтів до особистого досвіду та творчої вигадки [4].

Основна мета Medium полягає в створенні платформи, яка надає можливість людям з різними поглядами ділитися своїми ідеями та історіями. Це не лише простір для авторів, а й платформа для тих, хто прагне навчатися. Бачення Medium спрямоване на те, щоб цінність контенту була важливішою за його соціальну популярність, що робить цю платформу унікальною у порівнянні з традиційними соціальними медіа (Рисунок 2.8).

# • Medium

Рисунок 2.8 – Платформа для ведення блогу «Medium»

Щоб зареєструватися на Medium, спочатку необхідно перейти на офіційний сайт платформи. Для входу обрати один із запропонованих способів: через обліковий запис Facebook, Google або за допомогою електронної пошти. Якщо користувач обирає реєстрацію через облікові записи, необхідно надати Medium доступ до інформації та погодитися з умовами користування. Після цього система запропонує вказати електронну адресу, ім'я та бажане ім'я користувача, яке стане частиною профілю [5] (Рисунок 2.9).



Рисунок 2.9 – Реєстрація на платформі «Medium»

Після активації профілю, Medium попросить вибрати інтереси та запропонує публікації й авторів для підписки, які відповідають уподобанням користувача. Коли всі налаштування завершені, користувач потрапляє на головну сторінку зі створеним і готовим до використання профілем (Рисунок 2.10).

edium Q Search	🗹 Write 🚨 🍩
+ For you Following	Staff Picks
	🎒 Steve Yegge
Arunangshu Das	The Death of the Stubborn Developer
How Do You Monitor a Node is Application	6d ago
in Production?	in I THOUGHT THIS WAS WOR by Scott O
Node, is has become a popular choice for building scalable and	A Review: Publishing an Absolute Flop
high-performance applications. However, once your Node.js	Nov 22
Dec9 ₩ 5 $\bigcirc$ [] <sup>+</sup> …	Mt In The Medium Blog by Scott Lamb
	Best gifts for writers, from the Medium community
🚭 In Coding Beauty by Tari Ibaba	Dec 9
This new JavaScript operator is an absolute	See the full list
Say goodbye to try-catch	Recommended topics
	Programming Calif Ingravement
✦ Sep 18 ♥ 5.8K ♥ 84 () ↓ ····	Programming Self Improvement

Рисунок 2.10 – Головна сторінка платформи «Medium»

Кожен користувач платформи має можливість публікувати статті на широкий спектр тем, включаючи: технології, науку, бізнес, мистецтво, життя та багато інших. Одним із переваг платформи є її простий інтерфейс, який полегшує форматування та оформлення текстів, а також додавання фотографій та відео.

Щоб написати статтю на Medium, необхідно на головній сторінці натиснути кнопку «Write», яка відкриє сторінку редактора статей. У редакторі спочатку необхідно ввести заголовок статті, який буде відображати її основну тему або ідею (Рисунок 2.11).

Medium дозволяє легко форматувати текст за допомогою вбудованих інструментів: виділяти заголовки, підзаголовки, робити текст жирним або курсивом, додавати цитати та марковані списки. Також можна вставляти

зображення, відео, гіперпосилання чи навіть елементи коду, щоб зробити статтю більш інтерактивною та цікавою (Рисунок 2.12).

Medium Draft in Illyakomar Saved	Publish	•••	Ĵ	3
(+) Tell your story				

Рисунок 2.11 - Сторінка створення статті на платформі «Medium»

Medium Draft in Illyakomar Saved	Publish	•••	Ą	٢
Розробка вебзастосунку Розробка вебзастосунку системи дистанційного навчання				
😌 Головна сторінка	w [→			
P03999442144 Co,CVC29-465643223 31C2994				

Рисунок 2.12 - Можливості редактора на платформі «Medium»

Коли текст статті готовий, внизу сторінки можна додати теги до статті, які допоможуть іншим користувачам легше знайти публікацію за відповідними тематиками.

Для збільшення кількості читачів Medium пропонує прив'язати профіль в соціальних мережах, таких як Facebook та Twitter, до Medium, оскільки будьхто, хто підписується на автора в Twitter або Facebook і вже має профіль у Medium, автоматично підписується на автора і в Medium (Рисунок 2.13).



Рисунок 2.13 – Прив'язка профілю до соціальних мереж на платформі «Medium»

Меdium надає авторам широкі можливості для поширення свого контенту завдяки системі публікацій. Публікації – це простір на платформі, де кілька авторів можуть ділитися своїми роботами під єдиним брендом, об'єднані спільною темою чи інтересом. Це дає величезну перевагу новим авторам навіть якщо вони ще не мають власної аудиторії, розміщення статті у відомому виданні дозволяє миттєво отримати доступ до широкого кола читачів. Видання можуть охоплювати різні теми — від технологій та бізнесу до способу життя та здоров'я. Для авторів це означає шанс на швидке визнання та збільшення аудиторії [5].

Medium надає авторам прості й ефективні інструменти для відстеження успішності їхніх публікацій. У розділі «Статистика» можна побачити, скільки користувачів переглянули історії, прочитали їх повністю, а також загальний час, витрачений на читання (Рисунок 2.14).

	a a	5 Common Goblins Hid	Words With Li ing Inside	teral	8:12		al 🕈
COPUL	WORDS	🔶 - 7 min read - Jul 20			<	۲	₫
					Views People who visited yo	ur story's page	
Lifetime	odated hourly				386 Total Views	170 Member Vie	rws
					216 Non-member Views		
\$234	4.7K	2.9K	62%	~~			
Earnings	Views	Reads	Read ratio ()	Boosted (j)	Aug 03	Aug 05 Aug 07	Aug 09
					100		
Monthly							

Рисунок 2.14 – Перегляд статистики на платформі «Medium»

Medium надає авторам зручну і надійну систему монетизації, що дозволяє заробляти гроші на контенті з мінімальними зусиллями. Це дозволяє авторам зосередитися на творчості, а не на складних аспектах управління контентом та заробітку. Medium пропонує партнерську програму, до якої можна приєднатися після виконання кількох простих вимог. Однак, щоб мати можливість заробляти, необхідно стати платним користувачем Medium (Рисунок 2.15).



Рисунок 2.15 – Партнерська програма на платформі «Medium»

Партнерська програма дозволяє авторам монетизувати свої публікації, шляхом розміщення їх за платною стіною для преміум читачів. Заробіток залежить від часу, який читачі проводять за читанням статей, а також від взаємодії з контентом та відгуків.

Переваги користування [6]:

 Проста у користуванні. Меdium не вимагає жодних знань з програмування, він автоматично форматує статті таким чином, щоб вони мали чистий, зручний для читання макет.

 Система монетизації. Партнерська програма Medium дозволяє кожному автору увімкнути монетизацію всіх своїх статей. Оплата авторам здійснюється залежно від того, скільки часу читають їхні статті користувачі.

– Безкоштовна у використанні. Medium є безкоштовною платформою для авторів та користувачів.

Недоліки користування:

– Обмежений контроль над контентом. Користувачі мають обмежений контроль над контентом, який буде відображатися або монетизуватися. Medium має право видалити або вилучити ваш контент, якщо він порушує умови надання послуг.

– Обмежені можливості монетизації. Партнерська програма, яка дозволяє авторам заробляти гроші на своїх статтях доступна не всім. Щоб взяти участь у партнерській програмі, ви повинні бути прийняті, а ваш заробіток визначається поєднанням залучення читачів.

 Залежність від платформи. Якщо в алгоритмах або політиці платформи відбуваються зміни, це впливає на видимість або охоплення.

#### 2.3. Платформа для ведення блогу Ghost

Ghost – універсальна платформа для ведення блогу, яка підходить для професійних блогерів та творців контенту. Ghost дозволяє користувачам зосередитися на написанні текстів завдяки своєму витонченому і

мінімалістичному дизайну. Ghost був розроблений Джоном Ноланом у 2013 році. Інтуїтивно зрозумілий інтерфейс не тільки полегшує створення та публікацію контенту, але й надає додаткові можливості для оптимізації та кастомізації (Рисунок 2.16).



Рисунок 2.16 – Платформа для ведення блогу «Ghost»

Щоб зареєструватися на Ghost, спочатку потрібно перейти на офіційний сайт платформи. Після цього, обрати опцію початку роботи, яка зазвичай пропонує безкоштовний пробний період або платний тарифний план. Далі користувач має створити обліковий запис, вказавши електронну пошту, пароль та ім'я користувача. Після підтвердження електронної пошти система завершить процес реєстрації та надасть доступ до панелі керування блогом (Рисунок 2.17).



Рисунок 2.17 - Реєстрація на платформі «Ghost»

Ghost надає авторам один із найкращих текстових редакторів, що поєднує простоту та функціональність. Редактор забезпечує зручний та невідволікаючий інтерфейс, при цьому пропонуючи всі необхідні інструменти для створення якісного контенту. Однією з ключових переваг є можливість оптимізації метаданих публікації, що допомагає зробити статті більш ефективними з точки зору SEO (Рисунок 2.18).

c Posts Published 2	monetize your content, SEO, and more.	blish Post settings
	My name is Tiago Silva, and I am a content creator who has been sharing my	Post LPJ View post 3
	experience since 2020. My goal is to give you the best info to help you pick	substack-vs-mailerite
	the right tool for your newsletter, while I write about something I enjoy.	www.ibesizekkant/ion.com/subsizek-vs-mailenita/
		2010/07/07
		Publish date
	💉 Affiliate Disclaimer: This page contains affiliate links. My content	Tags
	is supported by readers like you. So if you buy after clicking on a link, I	Guiden ×
	get a commission without costing you extra. 🤕	Post access
		Public
		Excerpt
	Substack vs MailerLite Summary	
	The main difference is that MailerLite has much better customization and tools	
	to help grow a newsletter, while Substack has a free-forever plan and is better to	Authors
	publish a newsletter and blog using only 1 tool.	Tiago Silve X
		AND TRANSPORTED AND
	The main similarity is that both Substack and MailerLite are good tools for	Template
	beginners thanks to their affordable prices and ease of use.	Narrow Feature Image
	Mailarl ite is fantactic value due to its low prices and great tools for multishing a	
	newsletter. And if you have a blog/site using a tool like WordPress. MailerLite can	Feature this post
	do wonders due to its integrations.	
		O Post history
	Substack is a unique tool because you can use it to publish a blog, newsletter,	
	and and and another a summarity without any tablet of historical bases duties and	<> Code injection >

Рисунок 2.18 - Можливості редактора на платформі «Ghost»

Ghost підтримує різноманітні контентні блоки, що надають авторам гнучкість у форматуванні статей:

- Зображення та галереї для візуального оформлення.

– Markdown або HTML для більш точного редагування та налаштування контенту.

- Відео та аудіо блоки для інтерактивності.

– Закладки та Call-to-action для вбудовування посилань та залучення користувачів.

- Текстові блоки із розширеним форматуванням (заголовки, цитати, списки).

Особливістю Ghost є функція сніпетів – збереження часто використовуваних текстів або посилань, що дозволяє значно економити час при

створенні нових публікацій. Наприклад, можна зберегти фрагменти HTMLкоду для партнерських посилань чи шаблонний текст (Рисунок 2.19) [7].

Редактор Ghost також підтримує інтеграції з популярними платформами: YouTube, Twitter, Spotify, Vimeo та іншими, дозволяючи легко додавати інтерактивний контент у статті.



Рисунок 2.19 - Створення сніпетів на платформі «Ghost»

Ghost CMS пропонує широкий вибір привабливих та професійних тем для створення блогів. Проте важливо зазначити, що Ghost не є конструктором сайтів, тому можливості налаштування без знання коду досить обмежені. Щоб створити або повністю налаштувати власну тему, необхідно редагувати HTML та CSS безпосередньо, що робить платформу менш гнучкою для користувачів без технічних навичок [7].

З іншого боку, перевагою є те, що теми для Ghost зазвичай продаються як одноразова покупка, що робить їх значно доступнішими у порівнянні з підписками на теми для WordPress. Розробники Ghost поступово додають більше можливостей для налаштування, які не вимагають кодування. Багато сучасних тем Ghost дозволяють кастомізувати окремі елементи інтерфейсу (Рисунок 2.20). Головна мета – забезпечення швидкого й ефективного створення контенту для професійних авторів та видавців.



Рисунок 2.20 - Теми для платформи «Ghost»

Ghost також надає відмінні вбудовані інструменти для SEO, що робить платформу однією з найкращих для блогів та вебсайтів. SEO є важливим аспектом для досягнення високих позицій у пошукових системах, таких як Google, тому Ghost має всі необхідні функції для цього, без необхідності додаткових плагінів або знань SEO. Платформа дозволяє налаштовувати URLадреси, створювати XML-карти сайту для кращої індексації, використовувати kahohiчні теги для уникнення дублювання контенту. Крім того, Ghost підтримує додавання мета-описів, схеми розмітки для кращого розуміння контенту пошуковими системами, а також створення файлів гоbots.txt для керування доступом бота до сторінок (Рисунок 2.21).



Рисунок 2.21 - Вбудовані інструменти для SEO платформи «Ghost»

Основним способом монетизації на Ghost є підписка, коли читачі платять за доступ до контенту на сайті. Завдяки вбудованій підтримці підписок, Ghost є однією з найкращих платформ для такого виду монетизації. Окрім підписок, на Ghost також можна заробляти через афілійований маркетинг та показ реклами (Рисунок 2.22).



Рисунок 2.22 – Монетизація контенту на платформі «Ghost»

Ghost CMS чудово підходить для командної співпраці завдяки системі ролей з чотирма рівнями дозволів для користувачів [7] (Рисунок 2.23):

 Адміністратор. Має повний доступ до керування контентом, налаштуваннями та параметрами сайту.

Редактор. Може публікувати та редагувати будь-які статті, а також запрошувати авторів.

 Автор. Може створювати, редагувати та публікувати лише власні статті, але не має доступу до чужого контенту.

 Співавтор. Може створювати й редагувати свої статті, проте не має права публікувати їх.

Staff					Invite people
Tiago S	Silva — <b>Ov</b>	vner			
Administrators	Editors	Authors	Contributors	Invited	
			20		
		No adm	inistrators found		

Рисунок 2.23 - Командна співпраця на платформі «Ghost»

Переваги користування:

 Простий інтерфейс. Ghost має зручний інтерфейс, що дозволяє користувачам писати статті не відволікаючись.

 Теми. Широкий вибір тем, що налаштовуються дозволяючи користувачам персоналізувати свої веб-сайти відповідно до їхнього унікального бренду або стилю.

 Пошукова оптимізація. Ghost оснащений вбудованими функціями пошукової оптимізації, які допомагають користувачам покращити видимість свого вебсайту в рейтингах пошукових систем.

– Адаптивний дизайн. Ghost автоматично адаптується до різних пристроїв і розмірів екрану гарантуючи, що веб-сайти чудово працюватимуть;

 Управління контентом. Ghost має потужний редактор контенту з функцією попереднього перегляду в реальному часі, що дозволяє побачити, як виглядатиме контент, перш ніж буде опублікований.

– Редактор розмітки. Ghost використовує зручний редактор розмітки, який дозволяє авторам легко форматувати та стилізувати контент.

Недоліки користування:

– Обмежений вбудований функціонал. Порівняно з іншими платформами, Ghost пропонує менше вбудованих функцій і плагінів.

 Складний процес налаштування. Налаштування Ghost вимагає від користувачів самостійного керування хостингом, реєстрацією домену та конфігурацією.

 Менша підтримка спільноти. Платформа має обмежені можливості офіційної підтримки.

#### 2.4. Платформа для ведення блогу Tumblr

Tumblr – це платформа для мікроблогінгу, яка дозволяє користувачам публікувати короткі текстові повідомлення, зображення, відео та інший медіаконтент. Дописи в Tumblr можуть містити тексти, фотографії, відео, аудіо, прямі трансляції, тощо. Через фокус на мультимедії Tumblr часто використовують письменники, художники, музиканти та інші творчі люди, щоб ділитися своїми роботами (Рисунок 2.24) [8].



Рисунок 2.24 – Платформа для ведення блогу «Tumblr»

Щоб приєднатися до Tumblr, необхідно перейти на офіційну сторінку реєстрації платформи, де буде запропоновано створити акаунт за допомогою електронної пошти, акаунта Google, Apple ID або іншого затвердженого ідентифікатора. Створення облікового запису на Tumblr є безкоштовним і простим у налаштуванні. Всі пости на Tumblr є публічними і всі користувачі мають можливість стежити за авторами. На платформі немає опцій приватного профілю, як в Instagram або Facebook (Рисунок 2.25).



Рисунок 2.25 – Реєстрація на платформі «Tumblr»

Tumblr пропонує авторам простий та інтуїтивно зрозумілий редактор для поєднує зручність створення контенту, який використання та базову функціональність. Платформа надає можливість легко створювати пости різних типів, таких як текст, зображення, відео, аудіо та посилання, з мінімальними зусиллями. Ключовою перевагою Tumblr є його інтерфейс, який дозволяє користувачам швидко публікувати контент без необхідності складних налаштувань (Рисунок 2.26). Tumblr не має жорстких обмежень чи правил щодо типів контенту або дизайну сторінок, які розміщують цей контент, що робить його дуже різноманітним.



Рисунок 2.26 – Редактора на платформі «Tumblr»

Ще однією важливою особливістю є система тегів, що дозволяє публікаціям потрапляти у відповідні потоки інших користувачів. Це дає авторам можливість швидко охопити широку аудиторію без додаткових зусиль. Tumblr також дозволяє користувачам взаємодіяти з контентом через коментарі та лайки, що робить платформу можливою не тільки для публікацій, але і для взаємодії з іншими користувачами (Рисунок 2.27).

Content you see	Control what you see across Tumbir. To see changes, refresh the page, or restart Tumbir on your devices.		
	Filtered Tags <u>Help</u> Add a filter	Add	
		Remove	
		Remove	
		Remove	
	#a.o.e	Remove	

Рисунок 2.27 - Система тегів на платформі «Tumblr»

Tumblr дозволяє легко адаптувати вигляд блогу без необхідності володіння технічними навичками, що робить платформу доступною для широкої аудиторії. У редакторі можна змінювати зображення заголовка, шрифти, налаштовувати кольорову палітру, оновлювати заголовок та опис блогу. Tumblr також дозволяє створювати додаткові сторінки, наприклад «Про автора».

Для користувачів, які бажають зробити свій блог більш унікальним, платформа пропонує бібліотеку тем. Ці теми можна налаштувати, щоб надати блогу класичний вигляд із боковою панеллю та біографією автора. Користувачі можуть легко підключити знайдену тему. Крім того, Tumblr пропонує змінити кольорову палітру, обираючи з 12 доступних схем. Для тих, хто має навички програмування, документація з розробки тем допоможе створити повністю особливий дизайн [9] (Рисунок 2.28).



Рисунок 2.28 – Різноманітні безкоштовні теми для платформи «Tumblr»

Tumblr – це одна з найкращих соціальних платформ, яка дозволяє стежити за іншими користувачами, ставити вподобайки, коментувати та

ділитися дописами. Вірусний пост може принести тисячі переглядів та нових підписників. Користувачі можуть спілкуватися через Live Chat, як у Facebook, ділитися музикою, публікувати власні думки, а також відкривати популярні записи через пошук, схожий на функціонал Twitter. За допомогою поширення допису, є можливість розміщувати чужий контент із коментарем (Рисунок 2.29).



Рисунок 2.29 – Можливості платформи «Tumblr»

Tumblr також має покращене SEO, яке дозволяє залучати нових користувачів. Публікації індексуються пошуковими системами, такими як Google, а використання відповідних тегів може підвищити видимість як на Tumblr, так і в результатах пошуку. Функції, такі як репости та вподобання, генерують зворотні посилання на основний блог, що додатково покращує SEO.

Загалом Tumblr є безкоштовною платформою, що дозволяє користувачам створювати власні блоги, переглядати пости інших користувачів і користуватися всіма ключовими функціями без фінансового внеску. Основне джерело фінансування Tumblr – реклама. Користувачі мають можливість оплатити послугу для видалення реклами.

Переваги користування [10]:

– Проста у користуванні. Є вбудовані інструменти та теми, які користувач може використовувати для полегшення роботи.

– Безкоштовність. Однією з важливих переваг Tumblr є безкоштовний хостинг, користувачі можуть створити блог, не витрачаючи кошти;

– Пошукова оптимізація. Використання тегів у постах підвищує видимість як на пошуку Tumblr, так і на звичайних пошукових системах, сприяючи збільшенню переглядів.

Недоліки користування:

– Обмежені функції для бізнес-блогів. Tumblr може бути менш ідеальним вибором для бізнес-блогів порівняно з іншими платформами, такими як WordPress.

 Проблеми з конфіденційністю. Багато контенту доступно публічно за замовчуванням, що може порушити конфіденційність.

### 3. ТЕОРЕТИЧНА ЧАСТИНА

#### 3.1. Опис проектних рішень та інструментів розробки

За результатами аналізу платформ аналогічного призначення, для розробки платформи було обрано:

- сервіс для розробки інтерфейсів та прототипування Figma;

- мову програмування TypeScript;
- фреймворк Next.js;
- бібліотеку React;
- редактор коду Visual Studio Code;
- база даних MongoDB;
- система контролю версіями Git.

Figma – сервіс для розробки інтерфейсів, який широко використовується у сфері дизайну інтерфейсу користувача (UI) та користувацького досвіду (UX) (Рисунок 3.1). Figma забезпечує спільну платформу для дизайнерів, щоб вони могли створювати, ділитися своїми дизайн-проектами. Figma стала популярним вибором серед дизайнерських команд, завдяки інтуїтивно зрозумілому інтерфейсу та потужній функціональності [11].



Рисунок 3.1 – Сервіс для розробки інтерфейсів та прототипів «Figma»

Figma надає багато можливостей для дизайну інтерфейсів, включаючи інструменти векторного редагування, можливість створення прототипів та елементів дизайну. Вона підтримує створення прототипів взаємодії, де дизайнери можуть визначати зв'язки між різними екранами та імітувати взаємодію з користувачем.

Figma – це потужний інструмент, який широко використовують для прототипування в дизайні. Інтуїтивний інтерфейс та інтерактивні можливості дають змогу дизайнерам створювати прототипи, що відтворюють взаємодію користувачів та робочі процеси. За допомогою таких функцій, як інтерактивні області, анімації та переходи, можна наочно демонструвати концепцію інтерфейсу. Це дозволяє командам та зацікавленим сторонам переглядати прототипи та надавати зворотний зв'язок. Завдяки цим перевагам Figma є одним із провідних інструментів для прототипування (Рисунок 3.2).



Рисунок 3.2 - Створення прототипів в «Figma»

Ресурси Figma створюються спільнотою Figma або дизайнерськими агентствами і можуть бути завантажені та використані безкоштовно. Вони особливо корисні для дизайнерів, які є новачками у Figma. Використовуючи готові елементи дизайну, дизайнери можуть зосередитися на творчих аспектах
дизайну, а не на технічних деталях створення кожного окремого елемента [12] (Рисунок 3.3).



Рисунок 3.3 - Спільнота в «Figma»

Система дизайну у Figma дозволяє створити централізовану базу для всіх елементів оформлення. Це сприяє узгодженості в роботі команди та мінімізує розбіжності. Дизайнери можуть розробляти гнучкі, масштабовані компоненти, які легко використовувати повторно. Такий підхід значно спрощує процес оновлення та підтримки дизайн-системи. Учасники команди мають доступ до цих елементів, інтегруючи їх у свої проекти. Це забезпечує єдиний стиль та гармонійний вигляд продукту або бренду.

Figma забезпечує зручний контроль версій, який дозволяє дизайнерам відстежувати та управляти змінами в проектах. Завдяки автоматичному збереженню кожної внесеної зміни, користувачі можуть переглядати історію версій, порівнювати їх, та за потреби повертатися до попередніх станів. Це сприяє впорядкованому робочому процесу та дозволяє уникнути втрати важливих напрацювань.

Додатковою перевагою є можливість командної співпраці в реальному часі, при якій зберігається точна історія змін. Це гарантує, що всі учасники мають доступ до актуальних даних і працюють із найновішою версією дизайну. Завдяки таким можливостям Figma виступає надійним інструментом для управління проектами та забезпечення узгодженості в команді.

Figma – це багатофункціональний інструмент для створення ілюстрацій та іконок, який поєднує потужні можливості векторного редагування з інтуїтивним інтерфейсом.

При роботі з Figma не потрібно турбуватися про завантаження або встановлення будь-якого програмного забезпечення на комп'ютер. Все працює безпосередньо з вебсайту, що дозволяє працювати з будь-якого місця і в будьякий час.

Переваги Figma:

 Спільна робота в реальному часі. Користувачі можуть разом працювати над проектом: правити, ставити питання, коментувати розробку.

Анімації. Підтримка створення анімацій та переходів між екранами.

 Безкоштовна у використанні. Безкоштовна версія дозволяє працювати з 3 файлами одночасно.

– Зберігання. Figma працює у браузері та додатку, а всі файли знаходяться у хмарі.

Інтеграції. Можна інтегрувати з іншими інструментами та сервісами, такими як Slack або Jira.

Векторне редагування. Figma має можливості векторного редагування, що дозволяє створювати та редагувати складні фігури.

ТуреScript – це мова програмування, яка розширює JavaScript з додаванням статичної типізації (Рисунок 3.4). Варто зазначити, що в кінцевому підсумку TypeScript компілюється в JavaScript, що дозволяє виконувати готовий код на клієнті (через веб-браузер) або на сервері (через веб-сервер). Він був розроблений компанією Microsoft і вперше випущений у жовтні 2012 року.



Рисунок 3.4 – Мова програмування «TypeScript»

У наведеному нижче коді визначена функція add, яка приймає два параметри типу number та повертає результат також у вигляді числа. Завдяки строгій типізації TypeScript забезпечує, що тільки значення відповідного типу можуть бути передані у функцію (Рисунок 3.5).



Рисунок 3.5 – Приклад типізації «ТуреScript»

У прикладі виклику add(10, "20") TypeScript згенерує помилку, оскільки другий аргумент має тип string, а функція очікує обидва параметри типу number. Ця помилка дозволяє уникнути потенційних багів, які могли б виникнути при роботі з неправильними типами даних у JavaScript.

Такий підхід до розробки підвищує надійність та підтримуваність коду, що є однією з ключових переваг ТуреScript.

## Переваги ТуреScript [13]:

– Статична типізація. TypeScript має систему типізації, яка дозволяє вказувати типи змінних і параметрів функцій під час компіляції. Це дозволяє виявляти помилки, пов'язані з типами на ранніх стадіях, роблячи код більш надійним і менш схильним до помилок.

– Підтримка ES6. Підтримка сучасних функції JavaScript, що дозволяє розробникам писати чистіший код.

– Підтримка ООП. ТуреScript підтримує концепції ООП, такі як класи, інтерфейси та успадкування, що дозволяє створювати структурований та організований код.

 Організація коду. ТуреScript надає вбудовану підтримку модулів та просторів імен для кращої організації коду. Модулі дозволяють інкапсулювати код в окремі файли, що полегшує керування та підтримку.

Next.js – це фреймворк для розробки веб-додатків, який спрощує процес створення швидких інтерактивних додатків. Заснований на React (популярна JavaScript бібліотека). Next.js був розроблений компанією Vercel (раніше - ZEIT) у 2016 році. (Рисунок 3.6).



Рисунок 3.6 – Фреймворк «Next.js»

Ідея Next.js виникла з прагнення створити простий та ефективний інструмент для розробки веб-додатків, який автоматично вирішує проблеми такі як серверний та статичний рендеринг, код-сплітінг і маршрутизація.

Однією з ключових можливостей Next.js є рендеринг на стороні сервера, що значно покращує продуктивність та пошукову оптимізацію (SEO) веб-

додатків. Це дозволяє веб-додаткам завантажуватися швидше та займати кращі позиції в результатах пошуку [14].

Також у Next.js є підтримка статичної генерації. Ця технологія створює файли під час етапу збірки, що дає змогу заздалегідь підготувати їх для обслуговування, без необхідності чекати отримання даних. Після цього файли розміщуються на серверах по всьому світу, забезпечуючи максимально швидке завантаження.

Завдяки впровадженню Turbopack, який значно прискорює процес збірки, та вдосконаленим Server Actions для безпечної та ефективної роботи на серверній стороні, Next.js закріпив своє місце серед провідних інструментів веб-розробки. Крім того, функція автоматичного розділення коду дозволяє завантажувати лише той код, який потрібен для конкретної сторінки, що покращує швидкість завантаження та продуктивність системи. Така гнучкість робить Next.js чудовим вибором для проектів різного масштабу.

Next.js також підтримує статичну регенерацію, яка поєднує переваги статичної генерації з можливістю динамічно оновлювати елементи, які не були заздалегідь згенеровані або зазнали змін після початкового створення.

Next.js забезпечує готові інструменти JavaScript, створені на базі Rust, що сприяє прискоренню процесу розробки додатків, водночас зберігаючи високу якість роботи додатків. Поєднання високої продуктивності, швидкодії, гнучкості та сильної підтримки спільноти, робить Next.js одним із найпопулярніших виборів серед сучасних веб-розробників.

Next.js найкраще підходить для розробки:

– блогів;

- інтернет-магазинів;

– новин;

- інформаційних центрів;

- аукціонів;

- соціальних мереж.

Основні переваги Next.js [15]:

 Ефективність обслуговування. Логічна організація коду та інструменти автоматичної оптимізації спрощують обслуговування та зменшують проблеми з продуктивністю.

 Швидкість розробки. Швидкі цикли розробки завдяки простоті фреймворку та загальноприйнятій методології.

 Оптимізація зображень. Автоматична зміна розміру та оптимізація формату, що зменшує час завантаження сторінок.

 Пошукова оптимізація. Покращена пошукова оптимізація завдяки рендерингу на стороні сервера, що має вирішальне значення для покращення видимості в Інтернеті.

React – це JavaScript бібліотека, яка здатна створювати прості, швидкі та масштабовані веб-додатки. React була розроблена компанією Facebook та вперше з'явилася в травні 2013 року і зараз є однією з найпоширеніших фронтенд-бібліотек для веб-розробки (Рисунок 3.7).



Рисунок 3.7 – JavaScript бібліотека «React»

React.js був створений для вирішення проблем розробки складних користувацьких інтерфейсів. Його основна мета – надати спосіб ефективного оновлення та рендерингу компонентів, коли змінюються дані.

React використовує концепцію «Virtual DOM» (Document Object Model), що є спрощеним аналогом реального DOM. Завдяки цьому React може швидко та ефективно оновлювати тільки ті компоненти, які потребують змін, що позитивно впливає на загальну продуктивність (Рисунок 3.8). Для розробників, які прагнуть опанувати більш складні аспекти React, вивчення таких інструментів, як хуки, контекстний API та серверний рендеринг, може допомогти у створенні стабільних та масштабованих додатків.



Рисунок 3.8 – Основна концепція роботи «Virtual DOM»

Серверні компоненти в React – це новий підхід, який дозволяє виконувати рендеринг компонентів на сервері, що суттєво знижує навантаження на клієнтську сторону. Завдяки цьому сервер може передавати у браузер готовий НТМL, що прискорює початкове завантаження сторінки та покращує SEO. Такий підхід дозволяє обробляти важкі операції на сервері, залишаючи клієнту лише легкі задачі, що позитивно впливає на продуктивність додатка.

Однією з ключових переваг серверних компонентів є зменшення обсягу JavaScript, який потрібно завантажити та виконати на клієнті. Це значно покращує користувацький досвід, особливо на пристроях із обмеженими ресурсами. Окрім цього, серверні компоненти забезпечують більш ефективну інтеграцію з серверними даними, дозволяючи напряму отримувати їх із баз даних або API. Ця концепція вже інтегрована в React 18 та активно використовується у Next.js, що робить її потужним інструментом для створення сучасних вебдодатків.

Основні переваги React [16]:

 Багаторазові компоненти. React.js дозволяє розбити "користувацький інтерфейс" на менші, автономні компоненти, які можна повторно використовувати в різних частинах додатку.

– Віртуальний DOM. Використовуючи віртуальний DOM, React мінімізує прямі маніпуляції з реальним DOM, що призводить до підвищення продуктивності. Замість того, щоб перерендерити всю сторінку, React визначає конкретні компоненти, які потребують оновлення, і оновлює лише ці елементи.

– Компонентна архітектура. Дозволяє розробникам створювати складні користувацькі інтерфейси, розбиваючи їх на менші багаторазові компоненти.

– Серверний рендеринг. Завдяки SSR початковий рендеринг веб-сторінки відбувається на сервері, а не в браузері клієнта. Це покращує продуктивність веб-сайту, зменшуючи час початкового завантаження. Оскільки сервер надсилає клієнту повністю відрендереною HTML-сторінку.

– SEO. React.js має можливість рендерити "віртуальний DOM" на сервері та надсилати HTML клієнту, що покращує видимість сайту в пошукових системах.

Visual Studio Code – це безкоштовний текстовий редактор з відкритим вихідним кодом від Microsoft. VS Code доступний для Windows, Linux та macOS. Хоча редактор відносно легкий, він включає в себе деякі потужні функції, які зробили VS Code одним з найпопулярніших інструментів середовища розробки останнім часом (Рисунок 3.9).

Visual Studio Code вперше був представлений Microsoft у 2015 році. Розробка редактора велася з урахуванням потреб сучасних розробників, які шукали легкий, ефективний та розширюваний інструмент для написання коду. Основою для Visual Studio Code послужила технологія Electron, що дозволяє використовувати веб-технології для створення настільних додатків.



Рисунок 3.9 – Редактор коду «Visual Studio Code»

Редактор швидко набув популярності завдяки своєму зручному інтерфейсу, широкому набору функцій та можливостям розширення за допомогою сторонніх розширень. Регулярно виходять оновлення, в яких додаються нові функції та вдосконалення.

VS Code пропонує вбудовану інтеграцію з Git та інструмент для спільної роботи Live Share, що робить його ідеальним вибором для командних проектів. Завдяки інтеграції з Git розробники можуть відстежувати зміни та вирішувати конфлікти безпосередньо в редакторі, що значно спрощує робочий процес.

Live Share підвищує рівень співпраці, дозволяючи команді працювати над одним кодом у режимі реального часу, незалежно від їхнього місцезнаходження. Це поєднання інструментів забезпечує не лише високу індивідуальну продуктивність, але й ефективну взаємодію між членами команди.

VS Code пропонує підтримку таких інструментів, як Jupyter Notebooks, і розширень, розроблених для роботи з даними та машинним навчанням. Редактор дозволяє писати та виконувати код на Python, створювати візуалізації та запускати моделі машинного навчання безпосередньо в середовищі. Завдяки цим можливостям він є зручним варіантом для проектів, пов'язаних з аналізом даних.

Основні переваги Visual Studio Code [17]:

Підтримка мов. VS Code підтримує JavaScript, TypeScript, Python, C#, Java, Go, Ruby та інші.

 Intelli-Sense. Визначає, чи залишився якийсь фрагмент коду незавершеним.
 Крім того, загальний синтаксис змінних і оголошення змінних виконуються автоматично.

 Розширення. Якщо користувач хоче використовувати мову програмування, яка не підтримується він може її завантажити і використовувати, як розширення.

– Підтримка Git. Ресурси можна клонувати зі сховища GitHub або зберігати.

 Швидкі команди. Дозволяє швидко виконувати команди та отримувати доступ до різних функцій лише кількома натисканнями клавіш.

– Кросплатформенність. Підтримується на операційних системах Windows, Linux та macOs.

МопgoDB – це документно-орієнтована база даних, відома своєю масштабованістю та гнучкістю. Серед її основних переваг – відкритий вихідний код, підтримка різних операційних систем, висока доступність і швидкість обробки даних. Вона використовує формат BSON (Binary JSON) для зберігання даних, що сприяє ефективному зберіганню та пошуку інформації. MongoDB є надійним рішенням для компаній різних галузей, таких як Toyota, Uber, Bosch та Urban Outfitters (Рисунок 3.10).



Рисунок 3.10 – Документно-орієнтована база даних «MongoDB»

MongoDB класифікується як база даних NoSQL, представлена у 2007 році. На відміну від реляційних баз даних, які організовані у вигляді таблиць, MongoDB підтримує гнучкі формати даних. Це робить її ідеальним вибором для швидко змінюваних та складних вимог до даних [18].

Основні переваги документованого формату MongoDb (Рисунок 3.11): – Зручна адаптація до різних змін у проекті. Модель документів MongoDB особливо ефективна при роботі з динамічними структурами даних. Замість фіксованих схем, вона забезпечує можливість змін у реальному часі, що робить її чудовим рішенням для проєктів із динамічними вимогами.

– Спрощення розробки ПЗ. Документна модель забезпечує просте відображення між об'єктами додатка та базою даних, що робить робочий процес більш зручним. Завдяки цьому складні перетворення даних потрібні значно рідше, що зменшує витрати часу та зусиль.

 Ефективність обробки даних. MongoDB дозволяє значно швидше отримувати дані завдяки документованого формату. Багато запитів можна виконати за один раз, що усуває необхідність об'єднань таблиць, характерних для реляційних баз даних.  Оптимізація зберігання. Можливість групувати подібну інформацію в межах одного документа. Це підвищує швидкість доступу до даних та дозволяє ефективно використовувати ресурси для їх зберігання.

– Підтримка складних структур. MongoDB чудово справляється з вкладеними структурами, такими як масиви або інші документи. Це особливо актуально для даних, що мають вкладену структуру, наприклад: коментарів до постів чи списку адрес у профілі користувача.

```
> db.student.find().pretty()
{
         "_id" : ObjectId("60263475f19652db63812e98"),
         "StudentName" : "GeeksA",
"StudentId" : "g_f_g_1209",
         "Branch" : "CSE",
         "Address" : [
                  {
                           "permaAddress" : " XXXXXXXXXX,
                           "City" : "Delhi",
                           "PinCode" : 202333
                  },
                  {
                           "currAddress" : "PPPPPPPPPPP",
                           "City" : "Mumbai",
                           "PinCode" : 334509
                  }
         ]
}
>
```

Рисунок 3.11 – Приклад документованого формату даних «MongoDB»

Коли слід використовувати MongoDB:

– Для додатків, де важливо мати швидкий доступ до даних та високу продуктивність.

Якщо немає можливості призначити окремого адміністратора бази даних,
 MongoDB стане вигідним вибором завдяки своїй простоті у використанні та низьким експлуатаційним витратам.

У контексті роботи з хмарними середовищами варто використовувати
 MongoDB, оскільки її можливість розподіляти дані між кількома серверами
 забезпечує гармонійну інтеграцію та ефективність у розподіленій архітектурі.

 У проєктах з непередбачуваними структурами даних, MongoDB дозволяє швидко адаптуватися без потреби в складній підготовці, характерній для реляційних баз даних.

– Коли команди працюють за Agile-методологіями, оскільки її гнучка можливість дозволяє легко адаптуватися до змін, усуваючи потребу в детальному попередньому плануванні.

Багато компаній обирають NoSQL бази даних, для обробки великих обсягів даних або роботи в гнучких середовищах. MongoDB також добре працює з мікросервісами, що робить її популярним вибором для гнучких додатків [17].

Git – це найпопулярніша, з відкритим вихідним кодом, розподілена система контролю версій (DVCS), яка використовується для більш ефективного управління малих і великих проектів (Рисунок 3.12).



Рисунок 3.12 - Система керування версіями файлів «git»

Git дозволяє розробникам створювати та керувати віддаленими сховищами роботи, розміщеними в Інтернеті або на сервері. Можливість співпрацювати з іншими користувачами, отримуючи дані з сховищ, до яких забезпечено спільний доступ. Система контролю версій дозволяє розробнику з'ясувати, хто вносив зміни і які саме, після чого можна провести тести, виправити виявлені помилки і реалізувати необхідну функціональність. У разі будь-яких непередбачуваних обставин, код можна повернути до будь-якої з попередніх робочих версій, що дозволяє значно заощадити час та зусилля (Рисунок 3.13).



Рисунок 3.13 – Принцип роботи «git»

У Git є можливість виникнення конфліктів, коли система не може автоматично об'єднати зміни з різних гілок. Це зазвичай трапляється, коли два або більше розробників одночасно змінюють однакові частини коду або файли. Наприклад, якщо одна гілка змінює певну лінію коду, а інша – вносить свої зміни в ту ж саму лінію, Git не зможе визначити, чиї зміни повинні бути прийняті. Такі ситуації можуть виникнути і коли один розробник видаляє файл, а інший вносить зміни в цей файл. Git не може самостійно вирішити, яку зміну залишити, і тому конфлікт потребує ручного втручання. Розв'язання конфлікту починається з того, що необхідно відкрито переглянути файл, у якому виник конфлікт. Git позначає ці конфліктні ділянки маркерами, що дозволяє розробнику чітко бачити, де саме відбулися зміни. Після цього потрібно вручну вибрати, які зміни слід зберегти, а іноді й об'єднати зміни з обох гілок. Після редагування файлу необхідно додати його до індексу та виконати коміт (Рисунок 3.14).



Рисунок 3.14 - Вирішення конфлікту в «git»

Переваги Git [20]:

 Контроль версій. Однією з найбільших переваг використання системи контролю версій є можливість співпраці з іншими членами команди.

– Хостинг коду. Можливість розміщувати свої Git-репозиторії на GitHub, роблячи їх доступними для всіх, хто має відповідні дозволи.

 Рецензування коду. Коли член команди вносить зміни до кодової бази, інші члени команди можуть переглянути їх і надати свої відгуки до того, як код буде перенесено в основну гілку.

– Історія та аудит. За допомогою Git є можливість переглядати всю історію проекту, включаючи кожну зроблену зміну і хто її зробив.

 Відкликання змін. Якщо щось пішло не так, можна легко повернути зміни за допомогою Git.

#### 3.2. Опис обраного підходу до створення програмного забезпечення

В результаті розгляду та аналізу вище перелічених засобів та інструментів розробки, було вирішено обрати гнучку модель (Agile model), як підхід до розробки платформи для ведення блогу викладача.

Agile – методології управління проектами, яка використовує гнучкий та ітеративний підхід до розробки програмного забезпечення. Agile-модель спроектована для того, щоб реагувати на зміни у вимогах та умовах ринку, прискорюючи процес розробки та полегшуючи співпрацю між розробниками та клієнтами.

Методологія Agile виникла завдяки групі менеджерів проєктів у сфері розробки програмного забезпечення. З часом вона здобула популярність у розробці програмного забезпечення, але також поширилася на багато інших галузей. Серед них фінанси, інформаційні технології, бізнес, мода, біотехнології та навіть будівництво.

Agile-модель варто використовувати, коли проект має змінні або неясні вимоги, непередбачувані виклики або необхідність швидкої адаптації. Це підходить для проектів, де часто змінюються умови, наприклад, у розробці програмного забезпечення, створенні нових продуктів або проектів з високим рівнем невизначеності.

Аgile-модель поділяє завдання на часові рамки, щоб забезпечити конкретну функціональність для релізу. Кожна збірка є інкрементною з точки зору функціональності, а фінальна збірка містить всі атрибути. Поділ всього проекту на невеликі частини допомагає мінімізувати проектний ризик і загальний час реалізації проекту [21] (Рисунок 3.15).



Рисунок 3.15 – Методологія управління проектами «Agile»

Чотири основні цінності Agile [22]:

– Люди, а не процеси та інструменти. Agile команди цінують командну співпрацю та командну роботу, а не самостійну роботу та роботу "за правилами".

– Робоче програмне забезпечення, а не документація. Програмне забезпечення, яке розробляють Agile команди, має працювати. Додаткова робота, така як документація, не настільки важлива, як розробка програмного забезпечення.

 Співпраця з клієнтами під час переговорів. Клієнти є надзвичайно важливими в рамках методології Agile. Agile команди дозволяють клієнтам вказувати, куди має рухатися програмне забезпечення.

 Реагування на зміни за планом. Однією з головних переваг Agile управління проектами є те, що воно дозволяє командам швидко змінювати стратегії та робочі процес.

- Основні переваги використання Agile [22]:

– Спільна командна робота. Agile сприяє безпосередньому спілкуванню між членами команди, що покращує взаємодію та зменшує ймовірність суперечок.

– Гнучкість адаптації до змін. Agile дозволяє швидко адаптувати стратегії проекту, що важливо у світі постійно змінюваних вимог до програмного забезпечення.

 Орієнтованість на потреби клієнтів. Agile сприяє активній співпраці з клієнтами, отримуючи їхній зворотний зв'язок для визначення пріоритетів, що дозволяє командам активно реагувати на зміни.

#### 3.3. Прототип інтерфейсу платформи

1. Перегляд головної сторінки (Рисунок 3.16). У верхній частині головної сторінки розташована шапка блогу, яка містить:

– Логотип представлений у вигляді графічного елемента, який слугує для ідентифікації блогу та забезпечує перехід на головну сторінку при натисканні.

– Навігаційна панель, яка містить посилання на розділи блогу, такі як: «Головна», «Статті», «Книги», «Про автора». Це забезпечує користувачеві швидкий доступ до потрібного розділу.

 Кнопка для зміни теми. Вона дозволяє перемикатися між світлою та темною темами оформлення інтерфейс.

Нижче шапки розміщено блок з інформацією про автора, що містить короткий текстовий опис, який представляє автора блогу (ім'я, наукова посада, соціальні мережі та зображення).

У центральній частині сторінки відображається список доступних статей. Кожна стаття представлена у вигляді картки, яка містить: назву статті, категорію (до якої належить стаття), дату написання та зображення.

🗵 Teacher Blog		Головна Статті	Книги Про автој	ра		*
		Ілля І висококвалі фізики та к	ГОРОВИЧ фікований фахівцем у га вантової механіки. © <b>1</b>	nysi meopen	ичної	
æ		(	2		Ł	
Платформи для оцінн студентів	овання знань	Платформи для оц студентів	цінювання знань	Пла сту	атформи для оцінюван дентів	іня знань
Алгоритми	20.12.2023	Алгоритми	20.12.2023	Алг	оритми	20.12.2023

Рисунок 3.16 – Головна сторінка

2. Перегляд статті (Рисунок 3.17). У верхній частині сторінки розташовані основні метадані статті:

- Категорія статті, яка допомагає користувачу орієнтуватися в тематиці.

- Назва статті. Заголовок, який вказує на зміст матеріалу.

- Короткий опис статті, який стисло пояснює ключову ідею статті.

У центральній частині сторінки розміщений основний контент статті, який може включати: текст, зображення, відео, кодові блоки для демонстрації програмного коду, таблиці та цитати.



Штучний інтелект стає все більш необхідним інструментом у сфері освіти, зокрема для викладачів. Одним із ключових сервісів штучного інтелекту для

Рисунок 3.17 – Перегляд статті

3. Перегляд сторінки з книгами (Рисунок 3.18). Сторінка перегляду книг розроблена для зручного ознайомлення користувачів із доступними книгами. Кожна книга представлена у вигляді інформаційного блоку, що містить такі дані:

- Назва книги. Посилання, яке дозволяє перейти до повного тексту книги, завантажити її або відкрити на зовнішньому ресурсі.

– Автори. Перелік авторів книги, поданий у форматі П.І.Б.

 Короткий опис, який узагальнює основну ідею або зміст книги, допомагаючи користувачам швидко ознайомитися.

– Дата написання.

Книги
Особливості розробки web-застосунків для системи дистанційного навчання з допомогою бібліотеки React 2010-2023
2010-2023 ОП Кошова, ОО Черненко, ТВ Чілікіна, II Комар В роботі розглядаються особливості розробки системи дистанційного навчання для студентів закладів вищої освіти з використанням мови програмування TypeScript та бібліотеки React.
Особливості розробки web-застосунків для системи дистанційного навчання з допомогою бібліотеки React
20.10.2023
ОП Кошова, ОО Черненко, ТВ Чілікіна, II Комар
В роботі розглядаються особливості розробки системи дистанційного навчання для студентів закладів вищої освіти з використанням мови програмування TypeScript та бібліотеки React.

Рисунок 3.18 - Сторінка з книгами

4. Перегляд інформації про автора (Рисунок 3.19). На сторінці перегляду інформації про автора надано детальні відомості, що дозволяють користувачу дізнатися більше про професійний досвід та наукові досягнення автора. Сторінка про автора може містити наступну інформацію:

 Публікації та дослідження автора. Основні наукові напрямки, в яких працює автор, а також перелік наукових публікацій, що свідчить про його внесок у розвиток певної галузі.

 Проєкти. Опис ключових наукових та інженерних проектів, над якими працював автор.

 Освіта. Відомості про академічні ступені, наукові спеціалізації та освітні установи, в яких автор здобув вищу освіту.

- Контактна інформація для комунікації та можливості зв'язку з автором.

🗵 Teacher B	
	Про автора
	Я є висококваліфікованим фахівцем у галузі теоретичної фізики та квантової механіки. Моя кар'єра охоплює дослідження явищ, пов'язаних з квантовою теорією поля. Я маю численні публікації у високорейтингових наукових журналах.

Рисунок 3.19 – Сторінка з інформацією про автора

5. Сторінка авторизації (Рисунок 3.20). Сторінка авторизації надає адміністратору можливість увійти в систему за допомогою наступних елементів:

- Текстові поля для вводу даних (пошта та пароль).

– Кнопка «Увійти». Активує процес авторизації після введення коректних даних.

	Вхід	
8	email@gmail.com	
	Пароль	0
	Увійти	

Рисунок 3.20 – Сторінка авторизації

 6. Головна сторінка адміністративного інтерфейсу (Рисунок 3.21).
 Сторінка забезпечує можливість перегляду статистики, пошуку та додавання нових статей. Основні елементи сторінки:

 Поле для пошуку за ключовими словами, що дозволяє швидко знайти потрібну статтю або книгу.

- Графік, що показує динаміку кількості статей.

- Числова статистика переглядів блогу, статей та книг.
- Зручний віджет для керування останніми статтями.



Рисунок 3.21 – Головна сторінка адміністративного інтерфейсу

 Сторінка статті (Рисунок 3.22). Сторінка статей забезпечує користувачам можливість перегляду, пошуку та додавання нових статей.
 Основні елементи сторінки:

 Таблиця зі статтями, яка містить наступні стовпці: назва статті (посилання на статтю), короткий опис статті, дата написання та кнопки управління (перегляд, редагування та видалення).

 Поле для пошуку статей за ключовими словами або фільтрами, що дозволяє швидко знайти потрібну статтю.

- Кнопка «Додати статтю». Відкриває форму для створення нової статті.

Tea	Cher Blo	٥g	Статті (7)			Додати	статтн	⊖ ∘
â	Головна	>	Назва	Опис	Дата написання	Дії		
	Статті		Сервіси штучного інтелекту для викладачів	Ця стаття розглядає, чому	14.11.2023	0	0_	Û
	Надацитирация	Ś	Сервіси штучного інтелекту для викладачів	Ця стаття розглядає, чому	14.11.2023	0	0_	Û
0	палаштування		Сервіси штучного інтелекту для викладачів	Ця стаття розглядає, чому	14.11.2023	0	0_	Û
			Сервіси штучного інтелекту для викладачів	Ця стаття розглядає, чому	14.11.2023	0	0_	Û
			Сервіси штучного інтелекту для викладачів	Ця стаття розглядає, чому	14.11.2023	0	0_	Û
			Сервіси штучного інтелекту для викладачів	Ця стаття розглядає, чому	14.11.2023	0	0_	Û
			Сервіси штучного інтелекту для викладачів	Ця стаття розглядає, чому	14.11.2023	0	0_	Û

Рисунок 3.22 – Сторінка статті

8. Сторінка для додавання статті (Рисунок 3.23). Сторінка для додавання статті надає можливість вводити всю необхідну інформацію та додавати матеріали. Основні елементи сторінки:

- Назва статті. Текстове поле для введення назви.

- Категорія. Випадаючий список для вибору категорії.

– Зображення. Поле для завантаження зображення, яке автоматично відобразить вибране зображення.

- Опис. Текстове поле для введення короткого опису статті.

Матеріал. Текстове поле, де користувач може ввести основний контент статті
 з використанням розмітки Markdown. Розмітка дозволяє використовувати різні
 форматування тексту (жирний, курсив, списки, заголовки тощо) та вставляти
 медіа-матеріали (зображення або відео).

– Кнопка «Опублікувати». Кнопка для збереження та публікації статті після введення всіх обов'язкових полів.

				$\bigcirc$
Теа	icher Blo	¢	Додати статтю <sup>Назва</sup>	Додати статтю
	Головна	>	Категорія	
all	Статті	>	Виберіть категорію	
	Книги	>	Зображення	
$\odot$	Налаштування	>	висерлъ зоораження	
			Короткий опис статті	
			Матеріал	
			B I <del>S</del>   ≔ ≟=   ≝ >	

Рисунок 3.23 – Сторінка для додавання статті

9. Сторінка налаштування (Рисунок 3.24). Сторінка налаштувань дозволяє адміністратору блогу редагувати свої особисті дані. Основні елементи сторінки:

- Наукова посада. Текстове поле для введення або зміни наукової посади.

- Пошта. Поле для зміни або оновлення електронної пошти.

- Основна інформація. Поле для введення короткої інформації про автора.

 Посилання на соціальні мережі. Текстові поля для введення посилань на особисті сторінки в соціальних мережах.

– Додаткова інформація. Текстове поле для введення додаткових відомостей з використанням розмітки Markdown.

		÷
Teacher Blog	Налаштування <sup>Наукова посада</sup> Викладач	Зберегти
(п) Головна > ш) Статті >	nourra email@gmail.com	
<ul> <li>Ш Книги</li> <li>Налаштування</li> </ul>	Основна інформація	
	*	
	Посилання на соціальні мережі ftps://www.facebook.com/	
	Додаткова Інформація В І <del>S</del>   := 1= 1=	

Рисунок 3.24 - Сторінка налаштування

# 4. ПРАКТИЧНА ЧАСТИНА 4.1. Архітектура програмного забезпечення

Архітектура програмного забезпечення для блогу викладача складається з двох основних компонентів: адміністративного вебдодатку (Frontend та Backend) та публічного вебсайту (Frontend). Кожен з цих компонентів виконує свою конкретну роль у забезпеченні ефективного функціонування платформи, а їх взаємодія через чітко визначені інтерфейси гарантує стабільність і масштабованість системи. Розглянемо детальніше архітектуру кожного з компонентів.

Адміністративний інтерфейс блогу реалізовано на платформі Next.js, що дозволяє поєднати можливості серверного рендерингу (SSR) та клієнтського рендерингу для досягнення максимальної швидкості завантаження сторінок. Адміністративний вебдодаток складається з двох частин: Frontend та Backend.

Frontend частина надає викладачам можливість ефективно управляти контентом блогу. Серед реалізованих функцій: додавання нових статей, редагування існуючих, видалення матеріалів, а також перегляд статистики про відвідування та взаємодію користувачів з контентом. Крім того, Frontend забезпечує можливість управління книгами викладача, додавання нових видань, редагування їхніх деталей, а також відображення цієї інформації на публічному сайті. Оскільки в платформі використовується Next.js, при кожному оновленні або зміні сторінки користувач отримує швидке завантаження завдяки підтримці серверного рендерингу.

Васkend частина обробляє запити від публічного сайту та адміністративного інтерфейсу, забезпечуючи необхідну взаємодію з базою даних. Вона відповідає за аутентифікацію користувачів, перевірку прав доступу та зберігання змін у базі даних. Васkend використовує АРІ для обміну даними з Frontend та для управління статтями, книгами та інформації про автора. Він також відповідає за обробку запитів на додавання, редагування та видалення записів, а також збереження результатів у базі даних MongoDB. Публічний вебсайт блогу, створений на Next.js, дозволяє відвідувачам переглядати статті, книги викладача, а також іншу публічну інформацію. Платформа використовує технології статичної генерації сторінок (SSG) і серверного рендерингу (SSR), що гарантує високу швидкість завантаження сторінок і покращує досвід користувачів.

Сторінки статей генеруються на основі контенту, що зберігається в MongoDB, та доступні для перегляду без необхідності аутентифікації користувачів. Це дозволяє користувачам швидко знайти та прочитати статті викладача. Завдяки використанню SSG, сторінки генеруються під час розгортання і кешуються, що суттєво зменшує час їх завантаження.

База даних MongoDB служить центральною базою даних для збереження всього контенту блогу – статті, книги та інформація про автора. MongoDB є документно-орієнтованою базою даних, що забезпечує високу гнучкість у зберіганні різноманітних типів даних та легкість масштабування системи.

Система аутентифікації та авторизації. Використання JWT (JSON Web Token) забезпечує захист доступу до адміністративного інтерфейсу блогу, дозволяючи доступ тільки авторизованим користувачам для керування контентом блогу.

Архітектура програмного забезпечення для блогу викладача є потужним і гнучким рішенням, яке ефективно поєднує сучасні технології для управління контентом, високу швидкість завантаження сторінок та безпеку доступу. Використання Next.js як для публічного сайту, так і для адміністративного інтерфейсу, MongoDB для зберігання даних та JWT для аутентифікації забезпечує стабільну роботу платформи з можливістю масштабування в майбутньому. Це дозволяє зберігати високу ефективність системи при збільшенні обсягів контенту та кількості користувачів (Рисунок 4.1).



Рисунок 4.1 – Архітектура програмного забезпечення

#### 4.2. Опис діаграми роботи платформи та розгортання

Для опису роботи платформи для ведення блогу було вирішено побудувати діаграму Прецедентів (Рисунок 4.2).



Рисунок 4.2 – Діаграма прецедентів

## ПРЕЦЕДЕНТ: АВТОРИЗАЦІЯ.

Ектор: Адміністратор блогу.

Передумова: Адміністратор не авторизований в системі.

Післяумова: Адміністратор авторизований в системі.

Сценарій:

- 1. Адміністратор вводить свій етаіl та пароль.
- 2. Адміністратор натискає кнопку «Увійти».
- 3. Адміністратор авторизований.
  - 3.1. Включення «Перевірка облікових даних».

## ПРЕЦЕДЕНТ: КЕРУВАННЯ СТАТТЯМИ

Ектор: Адміністратор блогу.

Передумова: Адміністратор авторизований у системі.

Післяумова: Статті оновлюються відповідно до виконаних дій: додавання, редагування або видалення.

Сценарій:

- 1. Адміністратор переходить до вкладки «Статті».
- 2. На екрані з'являється список статей.
- 3. Адміністратор вибирає потрібну статтю.
- 4. Адміністратор оновив статтю.
  - 4.1. Включення «Додавання, перегляд, оновлення та видалення статей»

## ПРЕЦЕДЕНТ: КЕРУВАННЯ КНИГАМИ.

Ектор: Адміністратор блогу.

Передумова: Адміністратор авторизований у системі.

Післяумова: Книги оновлюються відповідно до виконаних дій: додавання, редагування або видалення.

Сценарій:

1. Адміністратор переходить до вкладки «Книги».

- 2. На екрані з'являється список книг.
- 3. Адміністратор вибирає потрібну книгу.
- 4. Адміністратор оновив книгу.
  - 4.1. Включення «Додавання, перегляд, оновлення та видалення книг»

## ПРЕЦЕДЕНТ: КЕРУВАННЯ НАЛАШТУВАННЯМИ.

Ектор: Адміністратор блогу.

Передумова: Адміністратор авторизований у системі.

Післяумова: Адміністратор оновив особисту інформацію про себе.

Сценарій:

- 1. Адміністратор переходить до вкладки «Налаштування».
- 2. Адміністратор переглянув можливості зміни профілю.
- 3. Адміністратор оновив особисту інформацію про себе.
  - 3.1. Включення «Зміна особистих даних»
  - 3.2. Включення «Додавання додаткової інформації про автора»

## ПРЕЦЕДЕНТ: ПЕРЕГЛЯД СТАТТІ

Ектор: Користувач.

Передумова: Користувач відкрив сторінку блогу

Післяумова: Користувач переглянув статті.

Сценарій:

- 1. Користувач переходить до вкладки «Статті».
- 2. На екрані з'являється список статей.
- 3. Користувач вибирає потрібну статтю.
- 4. Користувач переглянув статтю.

## ПРЕЦЕДЕНТ: ПЕРЕГЛЯД КНИГИ

Ектор: Користувач.

Передумова: Користувач відкрив сторінку блогу.

Післяумова: Користувач переглянув книгу. Сценарій:

- 1. Користувач переходить до вкладки «Книги».
- 2. На екрані з'являється список книг.
- 3. Користувач вибирає потрібну книгу.
- 4. Користувач переглянув книгу.
  - 4.1. Розширення «Завантаження книги».

#### ПРЕЦЕДЕНТ: ПЕРЕГЛЯД ІНФОРМАЦІЇ ПРО АВТОРА

Ектор: Користувач.

Передумова: Користувач відкрив сторінку про блогу.

Післяумова: Користувач переглянув інформацію про автора. Сценарій:

- 1. Користувач переходить до вкладки «Про автора».
- 2. На екрані з'являється інформація про автора.
- 3. Користувач переглянув інформацію.
  - 3.1. Розширення «Перехід на соціальні мережі».

Для розгортання проєкту було вирішено обрати платформу Vercel, оскільки вона забезпечує зручний та швидкий процес розгортання проекту. Її перевагами є автоматизація, підтримка CI/CD, простота інтеграції з репозиторіями та можливість налаштування змінних середовища. Це дозволяє розробникам мінімізувати час, витрачений на налаштування інфраструктури (Рисунок 4.3).



Рисунок 4.3 – Платформа для розгортання «Vercel»

Спочатку необхідно підготувати проєкт до розгортання. Для цього потрібно встановити всі залежності за допомогою «npm install», а також перевірити локальну роботу програми. Особливу увагу потрібно приділили файлу package.json, у якому потрібно вказати скрипт для збірки «build»: «next build» (Рисунок 4.4).

Q	EXPLORER		package.json ×
	$\sim$ open editors		i package.json > {} dependencies
Q	× 🕫 package.json		You, last week   1 author (You)
1	$\vee$ TEACHER-BLOG	다.다.이 @	1 { "norma", "toochon blog"
ይ	🗸 🔯 src		Z name : teacher-blog ,
ō	🗸 🋅 app		4 "private": true,
	> 🖿 books		Debug
±,≻	★ favicon.ico		5 "scripts": {
	🗦 globals.css		6 "dev": "next dev",
A-	🔮 layout.tsx		7 "build": "next build",
	🎡 loading.tsx		8 "start": "next start",
	🏘 page.tsx		9 "lint": "next lint"
QO			10 },

Рисунок 4.4 – Конфігураційний файл «package.json»

Наступним кроком потрібно створити акаунту на платформі Vercel та інтегрувати з репозиторієм проєкту через GitHub. У процесі підключення Vercel автоматично розпізнає технологію проєкту та налаштує базові параметри для збірки та розгортання. Перед запуском розгортання необхідно додати змінні .env,

що забезпечують збереження конфіденційних даних, таких як ключі доступу до АРІ (Рисунок 4.5).

<b>O</b> illyakomar/teacher-blog	9 main	
Choose where you want to creat	e the project and give it a name.	
Vercel Team	Project Name	
illyawise's projects	Hobby V teacher-blog-7c8m	
Framework Preset		
Next.js		$\sim$
Root Directory		
./		Edit
> Build and Output Settings	3	
✓ Environment Variables		
Key	Value i	
EXAMPLE_NAME	19JU23NF394R6HH	- ]
+ Add More		

Рисунок 4.5 – Створення проєкту для розгортання

Процес розгортання виконується автоматично після вибору репозиторію та натискання кнопки «Deploy». Платформа виконала збірку проєкту, а після успішного завершення надано посилання на готовий сайт. У разі виявлення помилок у роботі сайту їх виправлення у вихідному коді та збереження змін у

репозиторію спричиняє автоматичний запуск повторного розгортання (Рисунок 4.6).

TINDWAN	Notes Tan Ren Notes	Status Ready	Production  Current	Visit		÷
		Duration (				
(MAR)	Привіт! Я Ілля Ігорович макантичникати банках з суді пара насті Сілах за на сантантичника за настаруу агнах услужування, аспантичника паратичника паратичника настаруют со сантан.	40s (17h ago)				
	<ul> <li>S. 4.</li> </ul>	Domains 4				
		teacher-blo	g.vercel.app +1			
		↓ teacher-blo	g-git-main-illyawises-projects.vercel.app			
	Octanni ctatti	-o- teacher-blo	g-fp82h6t1u-illyawises-projects.vercel.app 🚯			
		Source				
		,7 main				
		-0- 054347c U	poare giobais.CSS			
eployment D	etails				Collaps	ISE
eployment D	etails				Collaps	ise
eployment D V Build Logs All Logs (57) (2) Error	etails			Q. Find in logs	Collaps 32s Ctrl F	se
eployment D V Build Logs All Logs (57) ③ Error 18:59:39.790	etails			Q, Find in logs	Collaps 32s Ctrl F	s
eployment D > Build Logs Allogs (37) (0) Em 18:59:39.790 18:59:39.791	etails ors (0) ( Warnings (0)  Compiled successfully Linting and checking valids:	ty of types		Q. Find in logs	Collaps 32s Ctrl F	se
eployment D V Build Logs All Logs () 18:59:39.790 18:59:39.791 18:59:45.440	etails ars(0) ( Warnings(0) < Compiled successfully Linting and checking validir Collecting page data	ty of types		Q. Find in logs	Collaps 32s Ctrl F	s
eployment D Build Logs All Logs (57) @ Em 18: 59: 39. 790 18: 59: 30. 791 18: 59: 44.440 18: 59: 47. 013	etails ors (0)	ty of types 8)		Q. Find in logs	Cottape 32s Ctrl F	s
Eployment D Build Logs (57) © Em 18:59:39.790 18:59:45.440 18:59:47.226	etails ors (0)	ty of types 8) 8)		Q. Find in logs	Collaps 32s Ctrl F	s
eployment D > Build Logs Allogs (37) (*) (*) 18:59:39.790 18:59:45.440 18:59:47.013 18:59:47.013 18:59:47.013 18:59:47.013	etails ors (0)	ty of types 8) 8)		Q. Find in logs	Collaps 32s Ctrl F	s
eployment D > Build Logs All Logs (*) 18:59:39.790 18:59:47.013 18:59:47.013 18:59:47.286 18:59:47.311 18:59:47.615 19:59:47.015 10:59:47.015 19:	etails or (0) ( Warnings(0) - Compiled successfully Linting and checking validir Collecting page data Generating static pages (2/) Generating static pages (4/) Generating static pages (4/) Generating static pages (6/)	ty of types 8) 8) 8)		Q. Find in logs	Collaps 32s Ctrl F	s
Build Logs     Build Logs     Build Logs     Build Logs     Build Logs     S9     S     S9     S     S     S	etails or:(0)	ty of types 8) 8) 8) 8)		Q, Find in logs	Collaps 32s Ctrl F	s

Рисунок 4.6 – Результат розгортання проєкту

Щоб вказати свій домен на Vercel, необхідно перейти до налаштувань проєкту в панелі керування Vercel. У розділі «Domains» потрібно додати домен: «https://teacher-blog.vercel.app/» (Рисунок 4.7). Проєкт готовий до використання.

General	Domains		
Domains Environments	These domains are as configured for each or	signed to your Production Deployments. Optionally, a different Git branch or a redirection to another domente.	iin can be
Environment Variables	Q mywebsite.com	· · · · · · · · · · · · · · · · · · ·	Add
Git			
Integrations Deployment Protection	Domain	Q teacher-blog.verceLapp	×
Functions	Environment	Production	~
Data Cache Cron Jobs	Redirect to 3	No Redirect	~
og Drains	Git Branch 🕤	Q All Branches ~	
Security	_		
A share a stat	Remove	Cancel	Cou

Рисунок 4.7 – Додавання домену

#### 4.3. Інструкція з використання платформи

1. Початок роботи. Для перегляду блогу необхідно перейти за посиланням https://teacher-blog.vercel.app/. Щоб мати можливість редагувати контент блогу, необхідно увійти в систему з адміністративними правами. Необхідно перейти за посиланням https://teacher-blog-admin-dashboard.vercel.app/login та заповнити відповідні текстові поля для входу (пошта та пароль). Після введення даних потрібно натиснути кнопку «Увійти» для отримання доступу до платформи (Рисунок 4.8). Якщо під час авторизації користувач вводить невірні дані (пошта або пароль не співпадають із записами в базі даних), система виводить відповідне повідомлення «Неправильно введено пошту або пароль!» (Рисунок 4.9). Якщо одне з полів залишено порожнім, система виводить повідомлення «Введіть пароль або пошту!» (Рисунок 4.10).



## Рисунок 4.8 – Сторінка авторизації

	Вхід
	Адміністратор ологу Пошта
	i-koma3r@ukr.net Пароль
	••••
	Увійти
"Вчити себе самого – благородна справа, але ще більш благородна – вчити інших"	
Марк Твен	Неправильно введено пошту або пароль!

Рисунок 4.9 – Помилка авторизації

	Вхід
	Адміністратор блогу
	Пошта
	email@gmail.com
	Введіть пошту
	Пароль
	Введіть пароль
	Увійти
"Вчити себе самого – благородна справа, але ще більш благородна – вчити інших"	
Марк Твен	

Рисунок 4.10 – Обов'язкові поля для авторизації

2. Після успішної авторизації користувач потрапляє на головну сторінку, де має змогу переглянути статистику: кількість переглядів, кількість статей, кількість книг, а також графік, що показує динаміку кількості статей за останні 6 місяців. Крім того, на сторінці є зручний віджет для останніх статей, за допомогою якого можна відразу додавати або редагувати статті, а також перейти на сторінку конкретної статті для детального перегляду (Рисунок 4.11).

TEACHER BLOG Панель адміністратора						
85 Головна 179 Статті 179 Кинги	<	Привіт, Ілля ! 👏	Кількість переглядів () 1000 +1831% за останей місядь	Кількіст 7 +19% за ост	ть статей 📳	Кількістькниг 🖽 10 +2 за останній місяць
Налаштування		Статті Кільмість статей за останні 6 місяція			Octaenhi citatti         Beboro 2 citatë         Image: State of the state of	Додати и: Шлях до Оптимальних Рішень у Графах озробці програмного забезпечения: Що обником по нас: як числа формулоть наш світ ива: Від відкриттів до технологій
		Сер	Вер	Лис	🕑 Редагувати	

Рисунок 4.11 – Головна сторінка

3. Для створення статті необхідно вибрати вкладку «Статті» в навігаційній панелі. Після вибору вкладки користувач потрапляє на сторінку з
усіма статтями (Рисунок 4.12). На сторінці також доступна можливість пошуку статей за назвою, а для зручності перегляду передбачена пагінація.

TEACHER BLOG						*
Панель адміністратора	<	Головна / Статті				
🔐 Головна		Статті (7)				+ Додати
📳 Статті		dodabanna, bodai yoanna, badailenna				
ниги Книги		Пошук				
🕄 Налаштування		Назва	Опис	Дата написання	Дата оновлення	
		( Теккології ) Технології в IT: Тренди та Майбутні Перспективи	Світ IT розвивається з неймовірною швидкістю, і т	29.08.2024	18.11.2024	
		Курси Як обрати найкращі курси для програмування	Ця стаття розглядає, чому програмістам варто пр	30.08.2024	18.11.2024	
		Фізика Чому фізика важлива. Від відкриттів до технологій майбутнього	У цій статті ми розглянемо, як фізика не лише впли…	30.08.2024	18.11.2024	
		Математика Ватематика навколо нас: як числа формують наш світ	Математика — це не лише сухі формули та складні	01.09.2024	18.11.2024	
		Технології Як стати веб-розробником	Сьогодні веб-розробка є однією з найбільш затр…	03.11.2024	18.11.2024	
		Технології Нові технології в розробці програмного забезпечення: Що варто	IT-сфера швидко змінюється, і для того, щоб зали…	16.11.2024	18.11.2024	
					Попередня	Наступна

Рисунок 4.12 - Сторінка зі статтями

Далі потрібно натиснути кнопку «Додати». Після чого користувач потрапляє на сторінку, де потрібно заповнити обов'язкові поля: назва, опис, матеріал, зображення та вибрати категорію. Якщо форма буде відправлена без заповнення обов'язкових полів, з'явиться повідомлення про помилку, що поля є обов'язковими для заповнення (Рисунок 4.13). Після заповнення всіх полів потрібно натиснути кнопку «Опублікувати» (Рисунок 4.14). Успішне додавання автоматично перенаправляє користувача на сторінку зі списком статей, де новостворена стаття відображатиметься в таблиці.

Додати статтю	Опублікувати
Назва	
Назва статті	
ведіть назву	
Сатегорія	
Виберіть категорію 🗸	
иберіть категорію	
Іображення	
Выберите файл Файл не выбран	
Іодайте зображення	
Эпис	
Короткий опис статті	
Введіть опис	
Иатеріал	
◆ B I H S U ** 注 Ⅲ Ⅲ ∅ 図 Φ □	•



<b>Ј</b> одати статтю				Опублікувати
азва				
Алгоритм Дейкстри: Шлях до Оптимальних Рішень у Графа				
атегорія			tilta	
Алгоритми ~				
ображення				
Выберите файл Файл не выбран				
пис				
Апгоритм Дейкстри є одним із класичних апгоритмів для п задачах, таких як навігація, мережі зв'язку та оптимізація і Python.	эшуку найкоротших шляхі маршруту. У цій статті ми р	ів у графі з невід'ємниї розглянемо основи ал	ми вагами ребер. Він широко вик горитму Дейкстри, як він працює	ористовується в різних а також реалізацію на
атеріал				
				0
★ ▶ B I H 5 U 99 注 Ξ Ξ @ ⊠ ↓ C				

Рисунок 4.14 – Додати статтю

Для того, щоб переглянути створену статтю, потрібно натиснути кнопку «Переглянути» в колонці «Дії» (Рисунок 4.15). Після чого користувач потрапляє на сторінку, де може переглянути результат створеної статті (Рисунок 4.16).

Опис	Дата написання	Дата оновлення
Світ IT розвивається з неймовірною швидкістю, і тех	29.08.2024	18.11.2024
Ця стаття розглядає, чому програмістам варто прохо	30.08.2024	<b>Дії</b> 18.11.2( © Переглянути
У цій статті ми розглянемо, як фізика не лише вплива	30.08.2024	<sup>18.11.2</sup> <sup>18</sup> Оновити <sup>18.11.2</sup> <sup>1</sup> Видалити
Математика — це не лише сухі формули та складні рі…	01.09.2024	18.11.2024 ···
	• سر ۲۰۰۰	•

Рисунок 4.15 – Дії в таблиці статті



Рисунок 4.16 - Сторінка статті

Для того, щоб відредагувати статтю, потрібно натиснути кнопку «Редагувати» в колонці «Дії». Після чого користувач потрапляє на сторінку, де є можливість змінити поля: назву, опис, матеріал, зображення та категорію. Після зміни полів потрібно натиснути кнопку «Оновити» (Рисунок 4.17). Успішне редагування автоматично перенаправляє користувача на сторінку зі списком статей, де відредагована стаття відображатиметься в таблиці.

чедагувати статтю	Оновини
la38a	
Алгоритм Дейкстри: Шлях до Оптимальних Рішень у Графа	
атегорія	100 m
Алгоритми 🗸	and the second se
ображення	
Выберите файл Файл не выбран	
пис	
Алгоритм Дейкстри є одним із класичних алгоритмів для п задачах, таких як навігація, мережі зв'язку та оптимізація Python.	ошуку найкоротших шляхів у графі з невід'ємними вагами ребер. Він широко використовується в різних маршруту. У цій статті ми розглянемо основи алгоритму Дейкстри, як він працює, а також реалізацію на
	@ *

Рисунок 4.17 – Редагувати статтю

Для того, щоб видалити статтю, потрібно натиснути кнопку «Видалити» в колонці «Дії». Після чого з'являється модальне вікно з попередженням: «Ви дійсно бажаєте видалити ?» (Рисунок 4.18).

гії в IT: Тренди та Майбутні Перспективи		Світ IT розвиваєтьс	я з неймовірною швидкістю,
айкращі курси для програмуви Ви дійсно бажаєте видалити?		Х	є, чому програмістам варто г
а важлива: Від відкриттів до те татика навколо нас: як числа ф	Відмінити	видалити	лянемо, як фізика не лише ві е лише сухі формули та скла
і веб-розробником			обка є однією з найбільш затр
нології в розробці програмного забезпечення: Що варто вивчити в 202	24 році?	IT-сфера швидко зм	иінюється, і для того, щоб зал
гм Дейкстри: Шлях до Оптимальних Рішень у Графах		Алгоритм Дейкстри	1 є одним із класичних алгор

Рисунок 4.18 – Видалення статті

4. Для створення книги необхідно вибрати вкладку «Книги» в навігаційній панелі. Після чого користувач потрапляє на сторінку з усіма

книгами (Рисунок 4.19). На сторінці також доступна можливість пошуку книг за назвою, а для зручності перегляду передбачена пагінація.

TEACHER BLOG				*
Панель адміністратора	С Головна / Книги			
<b>В</b> Головна	Книги (2)			+ Додати
🕑 Статті	Madaaanni kada kaanni andaaanni			
🕮 Книги	Пошук			
🛱 Налаштування	Назва	Автори	Дата створення	
	Системи та технології в університеті	І.І. Комар, О.П. Кошова, Д.С. Тацій	03.11.2024	
	Алгоритми і структури даних: шлях до ефективного коду	1.1. Комар, О.П. Кошова	13.11.2024	

Рисунок 4.19 - Сторінка з книгами

Далі потрібно натиснути кнопку «Додати». Після чого користувач потрапляє на сторінку, де потрібно заповнити обов'язкові поля: назва, опис, вибрати файл у форматі PDF та додати П.І.Б авторів. По завершенні потрібно натиснути кнопку «Додати» (Рисунок 4.20). Успішне додавання автоматично перенаправляє користувача на сторінку зі списком книг, де новостворена книга відображатиметься в таблиці. Якщо форма буде відправлена без заповнення обов'язкових полів, з'явиться повідомлення про помилку, що поля є обов'язковими для заповнення. (Рисунок 4.21) Головна / Книги / Додати

<b>1одати книгу</b>		Додат
азва		
Системи та технології в ун	іверситеті	
айл		
Выберите файл Файл не	выбран 🛛 🛱 Пе	Іереглянути файл
пис		
пис Дізнайтесь, як правильно аспекти архітектури, вклю	проєктувати, розробляти та підтр чаючи принципи MVC	гримувати масштабовані веб-застосунки. Ця книга охоплює ключові
пис Дізнайтесь, як правильно аспекти архітектури, вклю Прізвище	проєктувати, розробляти та підтр наючи принципи MVC Ім'я	гримувати масштабовані веб-застосунки. Ця книга охоплює ключові По батькові
пис Дізнайтесь, як правильно аспекти архітектури, вклю Прізвище Комар	проєктувати, розробляти та підтр каючи принципи MVC Ім'я Ілля	гримувати масштабовані веб-застосунки. Ця книга охоплює ключові По батькові Ігорович

Рисунок 4.20 – Додати книгу

Додати книгу			Опублікувати
Назва			
Назва книги			
Введіть назву			
Файл			
Выберите файл Файл не выбр	ан		
Додайте PDF файл			
Опис			
Короткий опис книги			
Введіть опис			
Прізвище	Ім'я	По батькові	
Введіть прізвище	Введіть ім'я	Введіть по батькові	

Рисунок 4.21 – Обов'язкові поля для створення книги

Для того, щоб переглянути створену книгу, потрібно натиснути на назву книги. Після чого користувач потрапляє на нову сторінку, де може переглянути книгу або завантажити (Рисунок 4.22).



Рисунок 4.22 – Перегляд книги

5. Для налаштування профілю необхідно вибрати вкладку «Налаштування» в навігаційній панелі (Рисунок 4.23). На сторінці користувач може змінити відповідні поля: наукова посада, пошта, зображення, основну та додаткову інформацію, додати посилання на соціальні мережі.

TEACHER BLOG		
Панель адміністратора	С Головна / Налаштування профілю	
<b>С</b> оловна	Налаштурання профілю	
📳 Статті	Палаштування профило	рфињ
🕮 Книги	Наукова посада	
🕸 Налаштування	Викладач	
	Пошта	
	illya-komar@ukr.net	
	Зображення	
	Выберите файл Файл не выбран	
	Основна інформація	
	Висококваліфікованній фазівцеми у галузі теоретичної фізики та квантової механіки. За свою кар'єру активно досліджував явица, пов'язані з квантовою теорією поля та алгоритмами, та маю численні публікації у високорейтингових наукових журналах.	
	Посилання на соціальні мережі	
	Instagram	
	🚱 Facebook ~ https://www.facebook.com/	
	Додати Додаткова інформація ▲ ▲ В І Н 5 및 99 І≡ ΙΞ 52 Φ 💭	@ []
	1., # Про автора	
	3 Я є **високохваліфікованим фахівцем** у галузі теоретичної фізи Про автора	
	<ul> <li>1/ims//https://st3.depositohotos.com/9888809/12768/i/458/deposi</li> <li>7</li> <li># ocsita</li> <li>* "воскоковалфікованим фахівцем у галузі</li> <li>* "села</li> <li>* "доктор наук" у галузі теоретичної фізики</li> <li>* "тигістр фізики** з відпианос</li> <li>* "пигістр фізики** з відпианос</li> </ul>	
	11. 12. ен Дослідкення та публікації 13.	

Рисунок 4.23 – Налаштування профілю

Для того, щоб переглянути інформацію про автора, потрібно перейти на сторінку блогу та вибрати вкладку «Головна» (Рисунок 4.24). На сторінці буде відображатися стисла інформація про автора. Для перегляду додаткової інформації потрібно вибрати вкладку «Про автора» (Рисунок 4.25).



Рисунок 4.24 – Перегляд інформації про автора



Головна Статті

Про автора

Книги

### Про автора

Я є **висококваліфікованим фахівцем** у галузі теоретичної фізики та квантової механіки. Моя кар'єра охоплює дослідження явищ, пов'язаних з квантовою теорією поля. Я маю численні публікації у **високорейтингових наукових журналах**.



### Освіта

- Доктор наук у галузі теоретичної фізики
- Магістр фізики з відзнакою

#### Навички

- Моделювання: використовую чисельні методи для розв'язання складних фізичних задач.
- Програмування: володію Python, C++ та MATLAB для проведення симуляцій і аналізу даних.
- Математичні методи: спеціалізуюсь на використанні математичного апарату для дослідження квантових явищ.

#### Проекти

- Квантові обчислення: Розробка нових алгоритмів для квантових комп'ютерів.
- Теорія струн: Моделювання взаємодій на основі струнної теорії.

### Контактна інформація

- Електронна пошта: example@domain.com
- GitHub: https://github.com/username

© 2025 Wises. All rights reserved

Рисунок 4.25 – Перегляд додаткової інформації про автора

### ВИСНОВКИ

Розробка блог-платформи для освітнього процесу залишається актуальною і сьогодні. В останні роки блоги стали популярним засобом вираження ідей, обміну знаннями та спілкування з іншими людьми. Саме тому в рамках кваліфікаційної роботи ставилися задачі по розробці платформи для ведення блогу викладача. Робота пройшла апробацію, по ній було опубліковано статтю.

Основні результати роботи:

- визначено актуальність теми кваліфікаційної роботи;
- сформульовані основні вимоги до програмного забезпечення;
- проведено огляд переваг та недоліків існуючих платформ;
- виконано опис проектних рішень та інструментів до розробки програмного забезпечення;
- обрано методологію створення програмного забезпечення;
- розроблено прототип інтерфейсу платформи;
- описано архітектуру програмного забезпечення;
- побудовано діаграму прецедентів;
- розроблено блог-платформу;
- складено інструкцію з користуванням.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- What Is Blogger? Discover the Amazing Features and Benefits [Електронний pecypc]. – Режим доступу до ресурсу: URL: https://profiletree.com/what-isblogger/ – Назва з екрану.
- Blogger Огляд 2024 Чому це Ваш найкращий вибір [Електронний ресурс].
   Режим доступу до ресурсу: URL: https://www.websiteplanet.com/uk/websitebuilders/blogger/ – Назва з екрану.
- Advantages And Disadvantages of Blogger (Blogging Platform) [Електронний pecypc]. Режим доступу до pecypcy: URL: https://www.compromath.com/2017/08/advantages-and-disadvantages-of-blogger.html Назва з екрану.
- 4. What Is Medium.com? [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.bloggersgoto.com/what-is-medium-com/ Назва з екрану.
- 5. How to Write on Medium in 2024 (A Beginner's Guide) [Електронний ресурс].
   Режим доступу до ресурсу: URL: https://www.thesideblogger.com/how-to-start-writing-on-medium/ Назва з екрану.
- Benefits of Writing on Medium [Електронний ресурс]. Режим доступу до ресурсу: URL: https://bloggingguide.com/benefits-of-writing-on-medium/ Назва з екрану.
- Ghost CMS review: The best for content membership websites [Електронний pecypc]. – Режим доступу до ресурсу: URL: https://expressionbytes.com/ghostcms-review/ – Назва з екрану.
- Tumblr: What It Is and How to Join It [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.lifewire.com/what-is-tumblr-7094867 – Назва з екрану.
- 9. WordPress vs Tumblr (2024) Which One is The Best? [Електронний ресурс].
   Режим доступу до ресурсу: URL: https://www.elegantthemes.com/blog/wordpress/wordpress-vs-tumblr-comparison Назва з екрану.

- 10. 9 Tumblr Benefits Why You Should Use It (What Is It Good for?) [Електронний pecypc]. Режим доступу до pecypcy: URL: https://www.pitiya.com/tumblr-benefits-reasons-to-have-tumblr-blogs.html Назва з екрану.
- What Is Figma Used For? Unveil The Power Of Figma In 2024 [Електронний pecypc]. – Режим доступу до ресурсу: URL: https://themeselection.com/whatis-figma-used-for/ – Назва з екрану.
- 12. What is Figma? It's advantages and complete guide [Електронний ресурс]. Режим доступу до ресурсу: URL: https://intellipaat.com/blog/what-is-figma/ Назва з екрану.
- 13. What Is TypeScript? A Comprehensive Guide [Електронний ресурс]. Режим доступу до ресурсу: URL: https://kinsta.com/knowledgebase/what-is-typescript/ Назва з екрану.
- 14. What is Next.js? Overview of the Powerful React Framework Sanity [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.sanity.io/glossary/next-js Назва з екрану.
- 15. Pros and Cons of Next JS 2024 Updated Version [Електронний ресурс]. Режим доступу до ресурсу: URL: https://pagepro.co/blog/pros-and-cons-ofnextjs/ – Назва з екрану.
- 16. 9 Advantages of React.js: Why Choose It for Your Web Project? [Електронний<br/>pecypc]. Режим доступу до ресурсу: URL:<br/>https://www.sayonetech.com/blog/advantages-of-react-js/ Назва з екрану.
- 17. What is Visual Studio Code? Practical Guide(2023) [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.educba.com/what-is-visualstudio-code/ – Назва з екрану.
- 18. MongoDB for Data Management: Handle Big Data Easily [Електронний pecypc]. Режим доступу до pecypcy: URL: https://www.devoteam.com/expert-view/mongodb-for-data-management-handle-big-data-easily/ Назва з екрану.
- 19. Git Interview Questions [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.interviewbit.com/git-interview-questions/ Назва з екрану.

- 20. The Benefits of Using Version Control Systems Like Git in Software Development [Електронний ресурс]. – Режим доступу до ресурсу: URL: https://hojaleaks.com/the-benefits-of-using-version-control-systems-like-git-insoftware-development – Назва з екрану.
- 21. What Is Agile? And When to Use It [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.coursera.org/articles/what-is-agile-a-beginners-guide Назва з екрану.
- 22. What is Agile methodology? (A beginner's guide) [Електронний ресурс]. –
   Режим доступу до ресурсу: URL: https://asana.com/ru/resources/agile methodology Назва з екрану.

# додаток а. вихідні коди

## ЛОГІН ФОРМА

```
'use client';
```

```
export default function UserAuthForm() {
```

```
const form = useForm<LoginSchemaType>({
```

resolver: zodResolver(loginSchema),

});

```
const [loading, setLoading] = useState(false);
```

```
const { notifySuccess, notifyError } = useNotify();
```

```
const searchParams = useSearchParams();
```

```
const router = useRouter();
```

```
const onSubmit = async (data: LoginSchemaType) => {
```

```
setLoading(true);
```

```
const result = await signIn('credentials', { ...data, redirect: false });
```

```
if (result?.error) {
```

```
notifyError(ApiErrorMessageService.get(result.error as any));
```

```
setLoading(false);
```

return;

```
}
```

```
notifySuccess('Успішно');
```

```
const callbackUrl = searchParams.get('callbackUrl') || '/dashboard';
router.replace(callbackUrl);
```

};

```
return (
<Card className='w-full max-w-md'>
<CardHeader>
<CardTitle className='text-2xl'>Bxin</CardTitle>
```

```
<CardDescription>Адміністратор блогу</CardDescription>
</CardHeader>
<CardContent className='grid gap-4'>
<Form {...form}>
<form
onSubmit={form.handleSubmit(onSubmit)}
className='w-full space-y-2'
id='form-login'
>
<div className='grid gap-2'>
<FormField
control={form.control}
name='email'
render={({ field }) => (
<FormItem>
<FormLabel>Пошта</FormLabel>
<FormControl>
<Input
type='email'
placeholder='email@gmail.com'
disabled={loading}
{...field}
\geq
</FormControl>
<FormMessage />
</FormItem>
)}
/>
</div>
```

```
<div className='grid gap-2'>
```

```
<FormField
control={form.control}
name='password'
render={(\{ field \}) => (
<FormItem>
<FormLabel>Пароль</FormLabel>
<FormControl>
<Input type='password' disabled={loading} {...field} />
</FormControl>
<FormMessage />
</FormItem>
)}
/>
</div>
</form>
</Form>
</CardContent>
<CardFooter>
<Button disabled={loading} className='w-full' type='submit' form='form-login'>
Увійти
</Button>
</CardFooter>
</Card>
);
}
СТОРІНКА СТАТТІ
const breadcrumbItems = [
```

{ title: 'Головна', link: '/dashboard' },

{ title: 'CTaTTi', link: '/dashboard/articles' },

];

```
export default async function ArticlesPage() {
const session = await getServerSession(authOptions);
await connectDb();
const articles = await Article.find(
{ owner: session?.user.id },
{},
{ owner: 0 }
)
let dataArticle = JSON.parse(JSON.stringify(articles));
return (
<PageContainer>
<div className='space-y-4'>
<Breadcrumbs items={breadcrumbItems} />
<Articles data={dataArticle} />
</div>
</PageContainer>
);
}
'use client';
interface ArticleProps {
data: ArticleOutput[];
}
export const Articles: React.FC<ArticleProps> = ({ data }) => {
const router = useRouter();
return (
 <>
  <div className='flex items-start justify-between'>
  <Heading
   title={`CTaTTi (${data.length})`}
```

```
description='Додавання, редагування, видалення'
/>
<Button
className='text-xs md:text-sm'
onClick={() => router.push(`/dashboard/articles/new`)}
>
<Plus className='mr-2 h-4 w-4' /> Додати
</Button>
</div>
<Separator />
<DataTable searchKey='title' columns={columns} data={data} />
</>
```

# ТАБЛИЦЯ

```
'use client';
interface DataTableProps<TData, TValue> {
  columns: ColumnDef<TData, TValue>[];
  data: TData[];
  searchKey: string;
}
export function DataTable<TData, TValue>({
  columns,
  data,
  searchKey,
}: DataTableProps<TData, TValue>) {
  const table = useReactTable({
  data,
  columns,
```

```
getCoreRowModel: getCoreRowModel(),
getFilteredRowModel: getFilteredRowModel(),
});
return (<>
<Input
placeholder={`Пошук ...`}
value={(table.getColumn(searchKey)?.getFilterValue() as string) ?? "}
onChange={(event) =>
table.getColumn(searchKey)?.setFilterValue(event.target.value)
}
className='w-full md:max-w-sm'
\geq
<ScrollArea
              className='h-[calc(80vh-220px)]
                                                  rounded-md
                                                                 border
                                                                          md:h-
[calc(80dvh-200px)]'>
<Table className='relative'>
<TableHeader>
{table.getHeaderGroups().map((headerGroup) => (
<TableRow key={headerGroup.id}>
{headerGroup.headers.map((header) => {
return (
<TableHead key={header.id}>
{header.isPlaceholder
? null
: flexRender(header.column.columnDef.header, header.getContext())}
</TableHead>
);
})}
</TableRow>
))}
</TableHeader>
```

91

```
<TableBody>
```

```
{table.getRowModel().rows?.length ? (
```

```
table.getRowModel().rows.map((row) => (
```

```
<TableRow key={row.id} data-state={row.getIsSelected() && 'selected'}>
```

```
{row.getVisibleCells().map((cell) => (
```

```
<TableCell key={cell.id}>
```

```
{flexRender(cell.column.columnDef.cell, cell.getContext())}
```

```
</TableCell>
```

))}

```
</TableRow>
```

))

):(

<TableRow>

```
<TableCell colSpan={columns.length} className='h-24 text-center'>
```

Дані відсутні

```
</TableCell>
```

```
</TableRow>
```

)}

```
</TableBody>
```

</Table>

```
<ScrollBar orientation='horizontal' />
```

</ScrollArea>

```
<div className='flex items-center justify-end space-x-2 py-4'>
```

```
<div className='space-x-2'>
```

<Button

```
variant='outline'
```

size='sm'

onClick={() => table.previousPage()}

disabled={!table.getCanPreviousPage()}

>

```
Попередня
</Button>
<Button
variant='outline'
size='sm'
onClick={() => table.nextPage()}
disabled={!table.getCanNextPage()}
>
Наступна
</Button>
</div>
</div>
</>
);
}
ΦΟΡΜΑ CTATTI
'use client';
interface IProps {
data?: ArticleOutput;
}
const MarkdownEditor = dynamic(
() => import('@uiw/react-markdown-editor').then((mod) => mod.default),
{ ssr: false }
);
const ArticleForm = (props: IProps) => {
const { data } = props;
const { notifyError, notifySuccess } = useNotify();
const { data: session } = useSession();
```

```
const [loading, setLoading] = useState(false);
```

```
const img = data?.img
```

```
?`${process.env.NEXT_PUBLIC_URL}/api/file/${data?.img}`
```

: null;

```
const [previewImage, setPreviewImage] = useState<string | null>(() => img);
```

const router = useRouter();

```
const defaultValues = data ? data : defaultArticle;
```

```
const form = useForm<ArticleSchemaType>({
```

resolver: zodResolver(articleSchema),

defaultValues,

```
});
```

```
const title = data ? 'Редагувати статтю' : 'Додати статтю';
```

```
const action = data ? 'Оновини' : 'Опублікувати';
```

```
const handleFileChange = (e: ChangeEvent<HTMLInputElement>) => {
```

```
const file = e.target.files?.[0];
```

```
if (file) {
```

```
const reader = new FileReader();
```

```
reader.onloadend = () => {
```

setPreviewImage(reader.result as string);

};

```
reader.readAsDataURL(file);
```

} else {

setPreviewImage(null);

```
}};
```

```
const onSubmit = async (dataArticle: ArticleSchemaType) => {
```

try {

setLoading(true);

if (data) {

```
await ArticleApiService.updateOne(data._id, {
```

...dataArticle,

```
imageName: data.img,
owner: session?.user.id,
});
} else {
await ArticleApiService.createOne({
...dataArticle,
owner: session?.user.id,
});}
notifySuccess('Успішно');
router.push(`/dashboard/articles`);
router.refresh();
} catch (error: any) {
notifyError(ApiErrorMessageService.get(error.message));
} finally {
setLoading(false);
}};
return (
<div>
<Card className='max-w-screen-xl'>
<CardHeader className='flex flex-row items-center justify-between gap-1'>
<CardTitle>{title}</CardTitle>
<Button
disabled={loading}
className='ml-auto'
type='submit'
form='form-article'
>
{action}
</Button>
</CardHeader>
```

```
<CardContent>
<Form {...form}>
<form
onSubmit={form.handleSubmit(onSubmit)}
className='w-full'
id='form-article'
>
<div className='space-y-4'>
<div className='flex flex-col gap-10 sm:flex-row sm:items-center'>
<div className='space-y-4 grow-[2]'>
<FormField
control={form.control}
name='title'
render={({ field }) => (
<FormItem className='max-w-md'>
<FormLabel>Hазва</FormLabel>
<FormControl>
<Input
disabled={loading}
placeholder='Назва статті...'
{...field}
/>
</FormControl>
<FormMessage />
</FormItem>
)}
>
<FormField
control={form.control}
name='category'
```

```
render={({ field }) => (
<FormItem className='max-w-md'>
<FormLabel>Kateropiя</FormLabel>
<Select
disabled={loading}
onValueChange={field.onChange}
value={field.value}
defaultValue={field.value}
>
<FormControl>
<SelectTrigger>
<SelectValue
defaultValue={field.value}
placeholder='Виберіть категорію'
\geq
</SelectTrigger>
</FormControl>
<SelectContent>
{articleCategory.map((category) => (
<SelectItem key={category.id} value={category.name}>
{category.name}
</SelectItem>
))}
</SelectContent>
</Select>
<FormMessage />
</FormItem>
)}
\geq
<FormField
```

```
control={form.control}
name='img'
render={({ field: { value, onChange, ...fieldProps } }) => (
<FormItem className='max-w-md'>
<FormLabel>Зображення</FormLabel>
<FormControl>
<Input
{...fieldProps}
               file:text-primary
className='
                                  file:font-normal
                                                     file:border
                                                                   file:border-solid
file:border-gray-300 file:rounded-md'
disabled={loading}
type='file'
accept='image/*'
onChange=\{(event) => \{
onChange(event.target.files?.[0]);
handleFileChange(event);
}}
\geq
</FormControl>
<FormMessage />
</FormItem>
)}
\geq
</div>
<div className='grow-[1] mx-auto relative'>
{previewImage && (
<Image
width=\{250\}
height=\{200\}
objectFit='cover'
```

98

```
src={previewImage}
alt='Перегляд'
className='p-3 border border-gray-300'
>
)}
</div>
</div>
<FormField
control={form.control}
name='description'
render={({ field }) => (
<FormItem>
<FormLabel>Опис</FormLabel>
<FormControl>
<Textarea
className='resize-none'
placeholder='Короткий опис статті'
disabled={loading}
rows = \{5\}
{...field}
\geq
</FormControl>
<FormMessage />
</FormItem>
)}
\geq
<FormField
control={form.control}
name='material'
render={({ field }) => (
```

<FormItem>

<FormLabel>Matepiaл</FormLabel>

<FormControl>

<MarkdownEditor

visible

value={field.value}

onChange={(value) => field.onChange(value)}

minHeight='500px'

maxWidth='1100px'

/>

</FormControl>

<FormMessage />

</FormItem>

)}/>

</div>

</form>

</Form>

</CardContent>

</Card>

</div>

);};

export default ArticleForm;

# СТОРІНКА КНИГИ

const breadcrumbItems = [ { title: 'Головна', link: '/dashboard' }, { title: 'Книги', link: '/dashboard/books' }, ]; export default async function BooksPage() { const session = await getServerSession(authOptions);

```
await connectDb();
```

const books = await Book.find({ owner: session?.user.id }, { }, { owner: 0 });

```
let dataBook = JSON.parse(JSON.stringify(books));
```

return (

```
<PageContainer>
```

```
<div className='space-y-4'>
```

```
<Breadcrumbs items={breadcrumbItems} />
```

```
<Books data={dataBook} />
```

```
</div>
```

```
</PageContainer>
```

);

```
}
```