

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

КОРНІЄНКО КОСТЯНТИН КОСТЯНТИНОВИЧ

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій,
канд. техн. наук, доцент
_____ О.В. Зелінська
«_____» _____ 2024 р.

ОНЛАЙН-ПЛАТФОРМА ОБМІНУ ДОСВІДОМ ІТ-СПЕЦІАЛІСТІВ

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Потапова Н. А., доцент кафедри
інформаційних технологій,

(назва кафедри)

канд. екон. наук, доцент

Оцінка: _____ / _____ / _____
(бали за шкалою ЄКТС / за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця – 2024

АНОТАЦІЯ

Корнієнко К.К. Онлайн-платформа обміну досвіду ІТ-спеціалістів. Спеціальність 122 «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній (бакалаврській) роботі наведено результати розробки онлайн-платформи для ІТ-спеціалістів, оскільки кількість фахівців у цій галузі стрімко зростає, що призводить до збільшення попиту на форуми такого роду. Досліджені основні конкуренти та аналіз цільової області. Розглянуто інструменти та методи, які дають змогу створити такий додаток. Описано інтерфейс користувача та реалізацію клієнтської та серверної частин.

Ключові слова: онлайн-платформа, блог, React, Redux Toolkit, TypeScript, ІТ-спеціалісти.

57 с., 58 рис., 40 джерел.

ABSTRACT

Kornienko K.K. Online Platform for Sharing the Experience of IT Specialists. Specialty 122 "Computer Science". Vasyl Stus Donetsk National University, Vinnytsia, 2024.

The main goal of this qualification work is to develop an online platform for IT specialists, as the number of professionals in this field is rapidly increasing, leading to a higher demand for forums of this kind. The main competitors and the analysis of the target area were studied. The tools and methods that make it possible to create such an application are considered. The user interface and the implementation of the client and server parts are described.

Keywords: Online platform, Blog, React, Redux Toolkit, TypeScript, IT Specialists.

57 p., 58 figures, 40 sources.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ФУНКЦІОНУВАННЯ ОНЛАЙН-ПЛАТФОРМ.....	6
1.1 Розвиток та сучасні тенденції використання онлайн-платформ	6
1.2 Сутність та принципи функціонування онлайн-платформ.....	10
1.3 Класифікація та аналоги розробок онлайн-платформ.....	13
РОЗДІЛ 2. АНАЛІЗ ТЕХНОЛОГІЧНИХ РІШЕНЬ ТА ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ ОНЛАЙН-ПЛАТФОРМИ ОБМІНУ ДОСВІДУ ІТ-СПЕЦІАЛІСТІВ.....	18
2.1 Підходи та методи розробки онлайн-платформи	18
2.2 Модель та концепція функціонування онлайн-платформи.....	20
2.3 Вибір інструментів для розробки онлайн-платформи.....	24
РОЗДІЛ 3. РОЗРОБКА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ОНЛАЙН-ПЛАТФОРМИ ОБМІНУ ДОСВІДУ ІТ-СПЕЦІАЛІСТІВ	28
3.1. Дизайнерські рішення проєкту онлайн-платформи.....	28
3.2 Реалізація серверної частини онлайн-платформи.....	33
3.3 Реалізація клієнтської частини онлайн-платформи.....	35
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ	53
ДОДАТКИ	57

ВСТУП

Останнім часом в Україні спостерігається стрімкий розвиток ІТ-індустрії з різних напрямків, що привертає все більше уваги до себе. З кожним днем зростає кількість людей, що вирішують вивчати програмування та інші аспекти ІТ для реалізації себе в цій сфері. І хоча існує безліч ресурсів для вивчення, початківцям часто не вистачає досвідчених порад. Відповіді на їхні запитання можуть дати лише професіонали. Тут на допомогу приходять веб-форуми для спілкування та обміну досвідом. Це місце, де кожен може отримати відповіді на свої запитання, а також знайти корисні поради та відповіді на схожі питання інших користувачів.

Найбільшою перевагою таких форумів є можливість не лише отримати конкретну відповідь на свої запитання, а й поглибити свої знання в різних областях ІТ, порівняти підходи та вирішення проблем. У світі, де продуктивність грає ключову роль, форум стає ідеальним інструментом для навчання та вирішення проблем, навіть коли ви далеко від комп'ютера. Зростаючий попит на ІТ-спеціалістів не лише в Україні, а й у всьому світі, робить цю сферу особливо привабливою для новачків, а форуми стають важливим інструментом для їхнього розвитку та інтеграції в галузь.

Ринок ІТ-спеціалістів постійно розвивається, вимагаючи все більшої кількості фахівців з різноманітними навичками. З пандемією Covid-19 багато компаній перейшли на дистанційну роботу, зробивши ІТ-сферу однією з найбільш затребуваних. Це стимулює багатьох людей до вивчення програмування, тестування та інших аспектів ІТ. Проте, процес становлення розробником забирає час і потребує належної підтримки, яку можуть забезпечити спеціалізовані форуми. [1]

Мета дослідження полягає у створенні та реалізації проєкту онлайн-платформи для обміну досвідом ІТ-спеціалістів на основі сучасних технологічних програмних та дизайнерських рішень.

Об'єктом дослідження є процес розробки онлайн-платформи обміну

досвідом ІТ-спеціалістів, який реалізує функціональні особливості комунікації спеціалістів, а також налагоджує співпрацю в даному середовищі.

Предметом дослідження є технології, інструменти та дизайнерські рішення щодо розробки онлайн-платформи обміну досвідом ІТ-спеціалістів.

Завданнями дослідження стали:

1. Проведення аналізу та оцінки сучасних тенденцій у використанні онлайн-платформ, що дає змогу виявити основні елементи передумов створення даного проєкту.
2. Огляд існуючих блогів для ІТ-спеціалістів, виявлення їх переваг і недоліків.
3. Дослідження та застосування спеціалізованого програмного забезпечення для розробки онлайн-платформи обміну досвідом ІТ-спеціалістів, виявлення його специфічних особливостей при побудові архітектури додатку та формуванні функціональних вимог.
4. Розробка та реалізація проєкту онлайн-платформи, що дозволить ІТ-спеціалістам ділитись досвідом з початківцями у сфері ІТ.

Робота складається зі вступу, трьох розділів, висновків, списку використаних посилань. В першому розділі висвітлено питання теоретичних засад функціонування онлайн-платформ. В другому розділі наведено результати аналізу технологічних рішень та інструментів для розробки онлайн-платформи обміну досвідом ІТ-спеціалістів. Третій розділ містить результати програмної реалізації онлайн-платформи.

Результати роботи були апробовані на I Міжнародній науково-практичній конференції «Прикладні аспекти сучасних міждисциплінарних досліджень» (18 листопада 2022 р., м. Вінниця) та IV Всеукраїнській науково-практичній конференції студентів аспірантів та молодих вчених «Прикладні інформаційні технології» (19 травня 2023 року, м. Вінниця).

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ОСНОВИ ФУНКЦІОНУВАННЯ ОНЛАЙН-ПЛАТФОРМ

1.1 Розвиток та сучасні тенденції використання онлайн-платформ

Започатковані як примітивні сторінки у Всесвітній павутині, онлайн-платформи вибудовували свій шлях від простих майданчиків для обміну інформацією до надзвичайно складних та інтегрованих цифрових екосистем. Їхні перші кроки відображали просту мету – забезпечити спільноту можливістю спілкування та обміну думками. Однак, з часом, ці платформи перетворилися на відомі гіганти, які об'єднали мільйони користувачів та стали важливою частиною цифрового ландшафту. Від самого початку до сьогодні, ця еволюція є вражаючим прикладом того, як швидкий технологічний розвиток може перетворити навіть найпростіші ідеї у великі та впливові ділові моделі.

Давайте розглянемо основні кроки розвитку, через які пройшли онлайн-платформи:

- Початки: Перші онлайн-платформи з'явилися в кінці 20-го століття, коли інтернет став доступним для широкої аудиторії. Ці ранні платформи, такі як форуми, дискусійні групи та онлайн-комуніті, були спрощеними, але значущими для обміну інформацією. [4]

- Зростання функціональності: З плином часу, онлайн-платформи почали розвиватися, додавати нові функції та сервіси. Наприклад, соціальні мережі почали з'являтися, дозволяючи користувачам обмінюватися не лише текстовою інформацією, але й фотографіями, відео та іншими мультимедійними контентом.

- Перехід до екосистем: Сучасні онлайн-платформи не просто надають послуги або функції, вони стали повноцінними цифровими екосистемами. Наприклад, платформи електронної комерції об'єднують продавців, покупців, логістичні компанії та інші учасники великої цифрової торговельної мережі. [40]

- Трансформація бізнес-моделей: Цей процес еволюції також призвів до

перегляду та трансформації бізнес-моделей. Компанії, що колись зосереджувалися на продажу продуктів або послуг, тепер звертають увагу на створення цифрових платформ та екосистем, які забезпечують цілковитий досвід для клієнтів.

- Майбутні перспективи: Хоча сучасні онлайн-платформи досягли великих успіхів, існують низка перспектив для подальшого розвитку. Зокрема, розвиток штучного інтелекту, розширена реальність та інші інновації можуть значно змінити спосіб, яким ми використовуємо та сприймаємо ці цифрові екосистеми. [41]

Далі поговоримо про зростання популярності електронної комерції. Електронна комерція зазнала значного розвитку з часу свого початку. Від перших експериментальних онлайн-магазинів і введення електронних платежів в Інтернеті до сьогодні, коли це стало нормою. Ми дослідимо цей шлях розвитку, щоб зрозуміти, як електронна комерція перетворилася з простого експерименту в найбільш популярну форму торгівлі. [18]

Технологічні інновації відіграли важливу роль у розвитку електронної комерції. Мобільні додатки, штучний інтелект та блокчейн змінили підходи до покупок та продажів в Інтернеті. Ці технології не лише розширюють можливості для споживачів і продавців, але і змінюють способи, якими ми сприймаємо процес електронної комерції.

Зручність та доступність є ключовими факторами, які призвели до популярності електронної комерції. З можливістю здійснювати покупки швидко та зручно, не покидаючи домів, онлайн-платформи стали важливою частиною життя багатьох людей.

Розвиток електронної комерції відкриває нові можливості для бізнесу. Компанії адаптуються до цього нового цифрового ландшафту, щоб привернути клієнтів і збільшити прибуток. [5]

Незважаючи на всі переваги, електронна комерція має свої виклики. Питання безпеки даних, конкуренції та регулювання потребують уваги. Однак, розвиток технологій та нові стратегії можуть допомогти подолати ці

розвитку. Онлайн-платформи відкривають доступ до навчальних ресурсів для широкого кола людей незалежно від їх місцезнаходження. Це особливо корисно для тих, хто не має доступу до традиційних освітніх закладів через географічні, фінансові або інші обмеження.

Онлайн-курси та тренінги дозволяють студентам вибирати теми, які вони хочуть вивчати, і працювати в своєму власному темпі. Це дозволяє кожному студентові зосередитися на своїх індивідуальних потребах та вибрати навчальну програму, яка відповідає їхнім цілям та інтересам. [7]

Онлайн-платформи пропонують широкий вибір курсів з різних галузей, від програмування та маркетингу до мистецтва та гуманітарних наук. Це дозволяє студентам вибирати саме ті теми, які їх цікавлять, та розвиватися в обраній галузі.

Деякі онлайн-платформи надають можливість взаємодії та співпраці між студентами та викладачами через форуми, чати та спільне завдання. Це стимулює обмін знаннями та досвідом між учасниками та сприяє більш глибокому розумінню матеріалу. [19]

Онлайн-платформи також стимулюють культуру постійного навчання, дозволяючи студентам отримувати нові знання та навички протягом всього життя. Вони надають можливість навчатися в будь-який час та в будь-якому місці, що допомагає підтримувати актуальність та конкурентоспроможність на ринку праці (рис 1.2).



Рисунок 1.2 – Графік кількості учнів онлайн платформ [3]

Далі розглянемо основні виклики та перспективи розвитку онлайн-платформ.

Збільшення обсягу особистих даних, що обробляються онлайн, створює загрозу їхнього незаконного доступу та порушення конфіденційності. Недостатність заходів безпеки може призвести до витоку даних та порушення приватності користувачів.

Розвиток онлайн-платформи також вимагає відповідного регулювання з боку уряду. Недостатнє регулювання може призвести до використання платформ для незаконних діяльностей, маніпулювання ринками або порушень прав споживачів. [20]

Питання етики використання онлайн-платформ включають в себе проблеми штучного інтелекту, алгоритмічної чесності та впливу на суспільство. Наприклад, алгоритми можуть виявити себе біасованими або сприяти розповсюдженню дезінформації.

Незважаючи на виклики, розвиток онлайн-платформ відкриває безліч можливостей для інновацій та розвитку. Нові технології, які виникають, можуть поліпшити безпеку даних, ефективність платформ та користувацький досвід, стимулюючи тим самим подальший розвиток і розширення онлайн-послуг.

1.2 Сутність та принципи функціонування онлайн-платформ

Визначення онлайн-платформ є ключовим аспектом розуміння їхнього функціонування та впливу на сучасне суспільство. Онлайн-платформи - це цифрові інфраструктури, які забезпечують можливість взаємодії між користувачами, споживачами та постачальниками послуг або товарів через Інтернет. Вони діють як посередники, об'єднуючи різні сторони ринку та надаючи їм зручний майданчик для взаємодії. [21]

Сутність онлайн-платформ полягає в їхній здатності створювати мережеві ефекти, розширювати масштаб та об'єм послуг або товарів, які можуть бути доступні для користувачів. Вони дозволяють сторонам знаходити

одна одну, взаємодіяти та здійснювати транзакції в цифровому середовищі.

У сучасному цифровому світі роль онлайн-платформ надзвичайно важлива. Вони стали невід'ємною частиною економіки та суспільства, сприяючи розвитку бізнесу, інновацій та спілкування. Їхнє значення постійно зростає, оскільки вони надають нові можливості для взаємодії та розвитку в різних сферах життя. [22]

Важливо відзначити, що онлайн-платформи можуть мати різні форми і функціональність. Деякі з них спеціалізуються на конкретних видовищах, таких як електронна комерція, соціальні мережі або освітні послуги, тоді як інші створюють універсальні майданчики, які об'єднують кілька видів діяльності. Наприклад, платформи, такі як Amazon або Alibaba, пропонують широкий спектр товарів для покупки, в той час як Facebook або LinkedIn забезпечують можливість спілкування та зв'язку між користувачами.

Далі поговоримо про принципи функціонування. Онлайн-платформи працюють на основі ряду ключових принципів, які дозволяють їм ефективно функціонувати та надавати значний вплив на сучасне суспільство. Один з найважливіших принципів - це мережевий ефект, що відображається у тому, що з кожним новим користувачем чи учасником платформи зростає її цінність для інших користувачів. Це створює позитивний зворотний ефект, який приваблює нових учасників та збільшує активність вже існуючих. [8]

Ще одним важливим принципом є двостороння ринкова платформа, яка сприяє взаємодії між двома або більше типами учасників, такими як покупці та продавці, роботодавці та робітники, або власники та орендатори. Цей принцип створює сприятливі умови для виникнення ефективної та стабільної екосистеми, де кожен учасник має свої інтереси та можливості.

Модель "постачальник-споживач" також грає важливу роль у функціонуванні онлайн-платформ, забезпечуючи простий та зручний механізм для здійснення транзакцій між постачальниками та споживачами товарів чи послуг. Ця модель може включати в себе різні види взаємодії, такі як продаж, оренда, обмін чи передача прав власності.

Усі ці принципи спільно створюють основу для ефективної роботи онлайн-платформ та їхнього впливу на сучасне суспільство. Вони допомагають створювати умови для зручної та ефективної взаємодії між учасниками та сприяють розвитку цифрової економіки.

Також торкнемось бізнес-моделі онлайн-платформ. Розуміння різних бізнес-моделей, які використовуються онлайн-платформами для генерації доходів, є ключовим аспектом їхнього успішного функціонування. Одна з найпоширеніших бізнес-моделей - це реклама, де платформа заробляє гроші на показах або кліках на рекламні банери, відеоролики чи текстові оголошення. Цей підхід особливо популярний у соціальних мережах та веб-сайтах з великим трафіком. [42]

Іншою поширеною бізнес-моделлю є комісійні з продажів, де платформа отримує певний відсоток від кожної угоди, яка була здійснена через неї. Цей підхід часто використовується в електронній комерції та на ринкових майданчиках, де платформа посередничає у торгівлі між продавцями та покупцями.

Бізнес-модель підписки дозволяє платформі стабілізувати дохід, забезпечуючи регулярний потік грошей від підписників, які оплачують за доступ до ексклюзивного або преміум контенту. Цей підхід особливо популярний серед медіа-платформ, таких як стрімінгові сервіси відео чи аудіо, онлайн-журнали та інші сервіси, що надають доступ до вмісту з високою вартістю.

Деякі платформи використовують бізнес-модель "плати за транзакцію", де користувачі сплачують за кожну конкретну дію або транзакцію, здійснену на платформі. Наприклад, це може бути плата за кожну покупку, резервацію або відгук. Цей підхід дозволяє платформі заробляти гроші від активної участі користувачів та стимулювати їх до здійснення більше транзакцій.

Кожна з цих бізнес-моделей має свої переваги та особливості і може бути вигідною залежно від характеру та мети платформи. Розуміння та використання відповідної бізнес-моделі є важливим елементом стратегії

розвитку онлайн-платформи.

1.3 Класифікація та аналоги розробок онлайн-платформ

Класифікація онлайн-платформ може бути проведена на основі різноманітних критеріїв, таких як функціональність, вид послуг, тип користувачів тощо. Давайте розглянемо кілька основних категорій онлайн-платформ:

- Ринкові майданчики (Marketplaces).

Це платформи, які забезпечують можливість обміну товарами або послугами між продавцями та покупцями. Ринкові майданчики можуть бути загальними (наприклад, Amazon, eBay) або спеціалізованими на певному виді товарів чи послуг. [23]

- Соціальні мережі (Social Networks).

Це платформи, які забезпечують можливість спілкування та обміну інформацією між користувачами. Вони дозволяють створювати особисті профілі, додавати друзів, публікувати контент тощо. Прикладами є Facebook, Instagram, LinkedIn. [39]

- Освітні та навчальні платформи (Educational Platforms).

Ці платформи надають доступ до навчальних матеріалів, курсів, лекцій тощо. Вони можуть бути спрямовані на самоосвіту або представляти спеціалізовані курси з певної галузі. Прикладами є Coursera, Udemy, Khan Academy. [24]

- Розважальні платформи (Entertainment Platforms).

Це платформи, які надають розважальний контент, такий як відео, музика, ігри, книги тощо. Вони можуть бути платними або безкоштовними для користувачів. Прикладами є Netflix, Spotify, Steam.

- Фінансові платформи (Financial Platforms).

Ці платформи надають послуги з фінансового характеру, такі як онлайн-банкінг, платіжні системи, інвестиційні платформи тощо. Прикладами є PayPal, Revolut, Robinhood.

- Технологічні платформи (Tech Platforms).

Це платформи, які надають доступ до технологій, інструментів розробки, хмарних сервісів тощо. Вони можуть бути використані для створення програмного забезпечення, веб-сайтів, мобільних додатків та іншого технічного вмісту. Прикладами є GitHub, AWS, Google Cloud. [25]

Тепер коли ми знаємо основні види онлайн платформ, давайте дослідимо які є аналоги в платформи, яку я маю реалізовувати, а саме платформи для обміну досвідом ІТ спеціалістів. [9]

Перша на черзі Stack Overflow. Це найбільша онлайн-спільнота програмістів, де користувачі можуть задавати та відповідати на питання з програмування, отримувати поради та обмінюватися досвідом. До переваг можна віднести широкий спектр тематик з програмування та розробки, велику спільнота експертів та активних учасників, швидку відповідь на питання та велику кількість рішень. Недоліками є те, що немає можливості дублювання питань і відповідей та присутня деяка складність у знаходженні точної відповіді через обсяг інформації (рис 1.3).

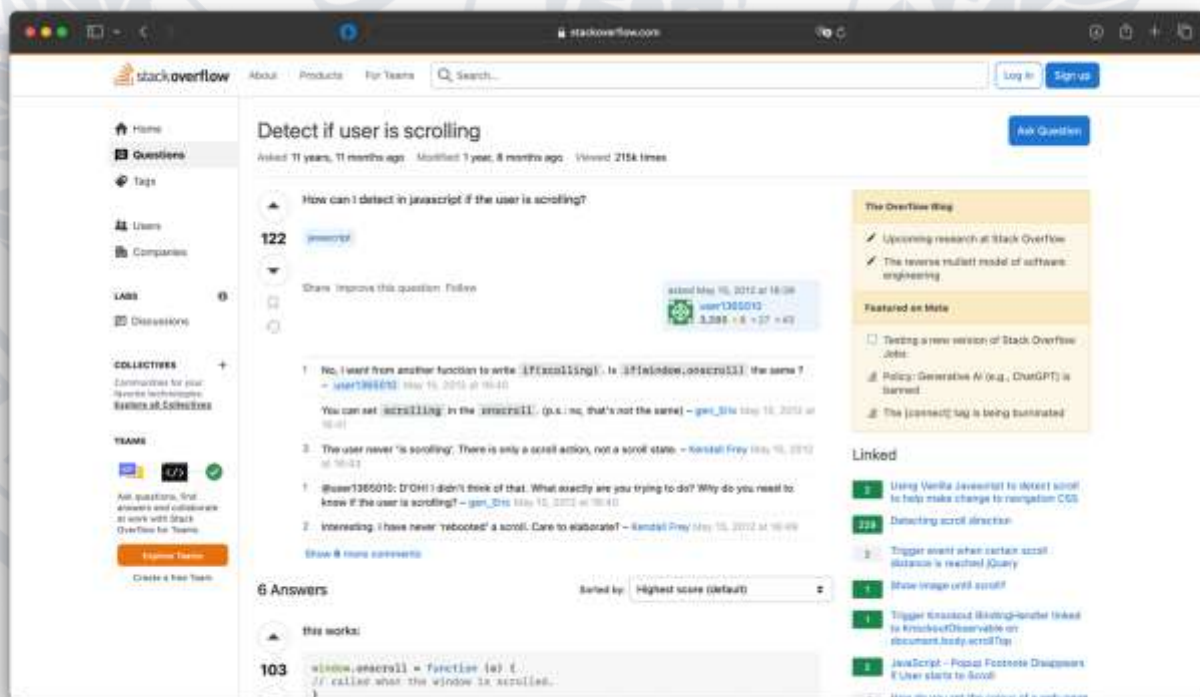


Рисунок 1.3 – Запитання про відстеження скролу на StackOverflow

Далі піде мова про Dev.to. Це спільнота розробників, де люди можуть публікувати свої статті, ділитися досвідом та навичками в галузі програмування та розробки. Серед переваг можна виділити можливість публікації статей для власного особистого брендування та професійного розвитку, функцію "Claps", яка дозволяє користувачам виражати свою підтримку та зацікавленість у контенті, а також різноманіття тем: від програмування до кар'єрних порад та особистого розвитку. [26]

Недоліками є можливість неперевіраних або неякісних матеріалів, оскільки публікувати статті може будь-хто, велика кількість контенту може призвести до важкості вибору та фільтрації релевантного матеріалу (рис 1.4).

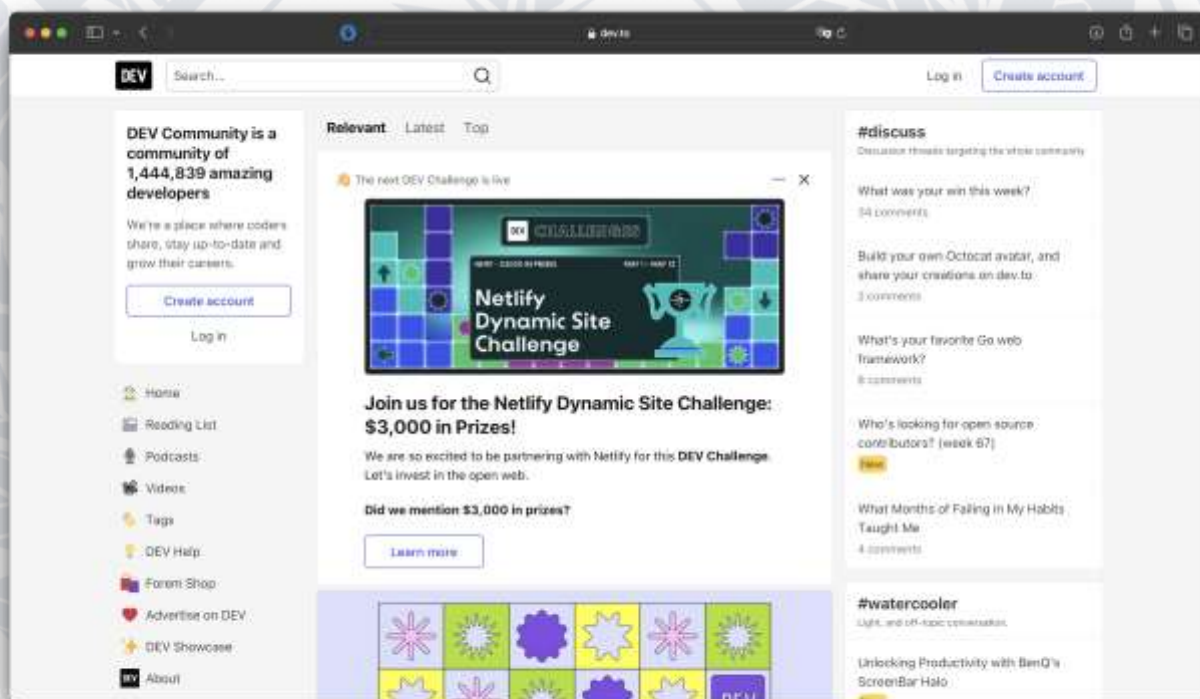


Рисунок 1.4 – Головна сторінка Dev.to

Наступна на черзі Medium. Це веб-сайт для публікації та читання статей на різні теми, включаючи програмування, розробку, архітектуру програмного забезпечення тощо. Перевагами є зручний інтерфейс та простота використання для публікації та читання статей, можливість реакції на контент у вигляді "claps" та коментарів, що сприяє взаємодії та обміну думками, широка

аудиторія читачів, що може сприяти популярності та розповсюдженню власного контенту. Серед недоліків можна виділити: обмежений доступ до аналітики для користувачів базового рівня, що може ускладнити відстеження ефективності власного контенту, відсутність можливості монетизації для більшості користувачів, які публікують контент (рис 1.5). [27]

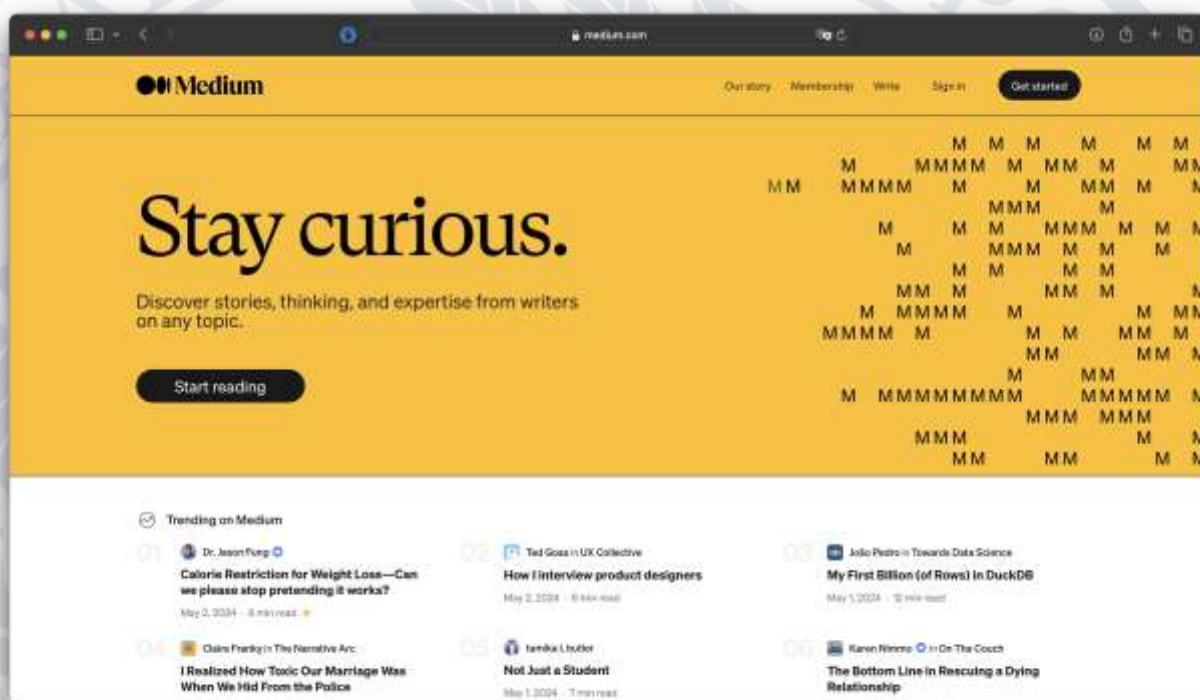


Рисунок 1.5 – Головна сторінка Medium

І остання це Hashnode. Спільнота веб-розробників, де можна публікувати статті, задавати питання, обговорювати теми з програмування та розробки веб-сайтів. Переваги: активна спільнота веб-розробників та ІТ-спеціалістів, яка ділиться досвідом та знаннями у сфері програмування та розробки, функціонал "Ask Me Anything" (AMA), що дозволяє користувачам задавати питання експертам та взаємодіяти з ними.

Недоліками є обмежені можливості монетизації контенту для користувачів, можливість надмірного спаму або неконструктивних коментарів у зв'язку з відкритою природою спільноти (рис 1.6).

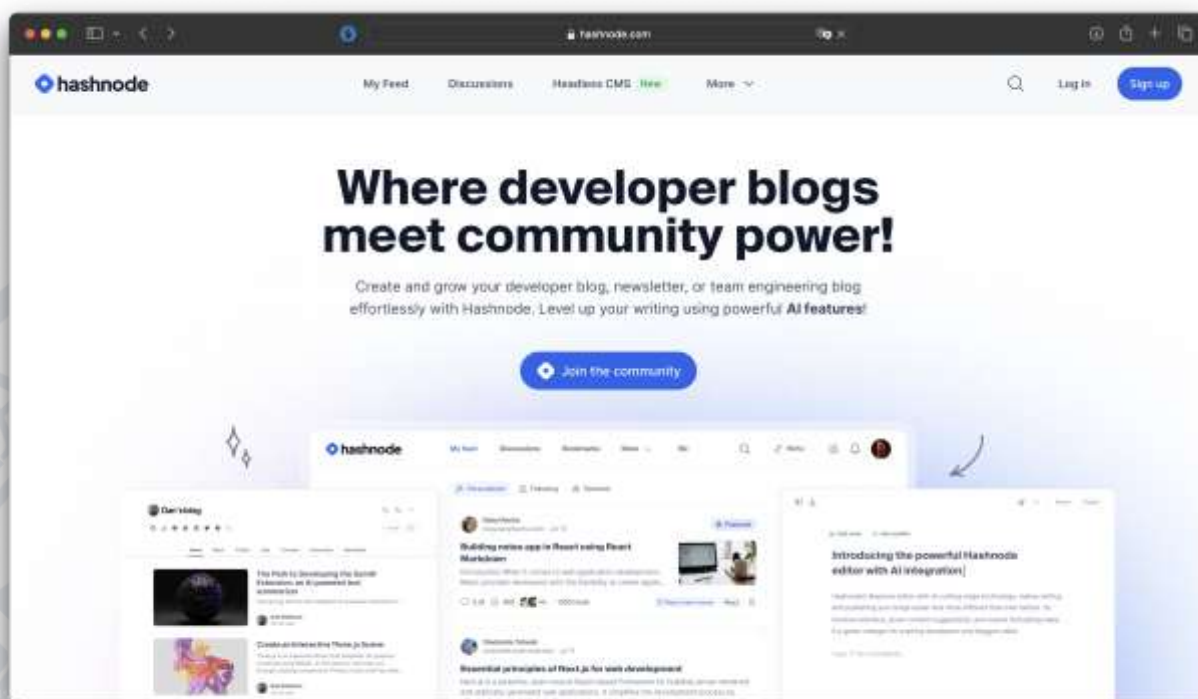


Рисунок 1.6 – Головна сторінка Hashnode

Отже, проведений аналіз онлайн-платформ спільнот, дозволив виявити основні комунікаційні характеристики та особливості контенту при обміні досвідом ІТ-спеціалістів. Серед них слід виділити такі складники як-от: формування власного контенту, залучення експертів, можливість надання оцінок (коментарі), монетизація контенту.

РОЗДІЛ 2.

АНАЛІЗ ТЕХНОЛОГІЧНИХ РІШЕНЬ ТА ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ ОНЛАЙН-ПЛАТФОРМИ ОБМІНУ ДОСВІДУ ІТ-СПЕЦІАЛІСТІВ

2.1 Підходи та методи розробки онлайн-платформи

Спочатку проведемо дослідження цільової аудиторії та конкурентного середовища. Перед створенням онлайн-блогу для ІТ-спеціалістів необхідно провести ретельне дослідження їх потреб та очікувань, а також вивчити конкурентне середовище. Дослідження цільової аудиторії включає опитування, спілкування та аналіз їхніх потреб, вимог та інтересів. Це дозволяє зрозуміти, які теми та формати контенту є найбільш цікавими для цієї аудиторії.

Паралельно з цим важливо аналізувати існуючі ресурси та блоги для ІТ-спеціалістів, щоб зрозуміти їх переваги та недоліки. Це допомагає виявити можливості для покращення та створення унікального продукту, який буде привабливим для аудиторії і конкурентоспроможним на ринку. [28]

Помітно, що успішність такого блогу буде залежати від його можливості відповісти на актуальні потреби та очікування аудиторії, а також вміння забезпечити унікальність і цінність пропонованого контенту. Тому важливо вивчати не лише сучасні тенденції, але й прогнозувати майбутні потреби ІТ-спеціалістів. Ретельне дослідження цільової аудиторії і конкурентного середовища є ключовим етапом у процесі створення блогу, що може визначити його подальший успіх на ринку обміну досвідом для ІТ-професіоналів.

Крім того, аналіз конкурентів допомагає визначити їхні сильні та слабкі сторони, а також виявити можливості для покращення власного продукту. Це може включати в себе розуміння того, які аспекти контенту або функціональності найбільш залучають користувачів, а також те, де самі конкуренти можуть бути недостатньо ефективними чи недостатньо відповідати потребам аудиторії. Такий аналіз допомагає створити стратегію

для розробки продукту, яка буде конкурентоспроможною та привабливою для користувачів.

Узагальнюючи, дослідження цільової аудиторії та конкурентного середовища - це ключові кроки у процесі розробки онлайн-платформи для обміну досвідом для ІТ-спеціалістів. Вони допомагають зрозуміти потреби та вимоги користувачів, а також визначити, яким чином можна відзначитися серед конкурентів. Це надає можливість створити унікальний продукт, який буде відповідати потребам аудиторії та привертати нових користувачів. [10]

Далі піде мова про методологію розробки. Методологія розробки онлайн-блогу для ІТ-спеціалістів - це систематичний підхід до процесу створення та вдосконалення цифрової платформи, спрямований на досягнення конкретних цілей та забезпечення високої якості продукту. Ця методологія може включати в себе різні етапи та кроки, а також використання різноманітних інструментів і технік розробки. [29]

Основні етапи методології розробки можуть включати збір вимог, аналіз конкурентного середовища, проектування інтерфейсу користувача, розробку та тестування програмного забезпечення, випробування на реальних користувачах, а також підтримку та післяпродажне обслуговування. [11]

Важливим аспектом методології є також визначення стратегії контенту та його постійне оновлення, щоб забезпечити актуальність та цінність для аудиторії. Крім того, важливо враховувати відгуки та реакції користувачів під час розробки та вдосконалення платформи, щоб забезпечити відповідність їхнім потребам та очікуванням.

Тестування та вдосконалення є важливим етапом в розробці будь-якого цифрового продукту, включаючи онлайн-блог для ІТ-спеціалістів. Цей процес дозволяє виявити помилки, недоліки та можливість поліпшення платформи перед її випуском для широкого користувача. Давайте розглянемо ключові аспекти тестування та вдосконалення:

Функціональне тестування: перевірка роботи різних функцій платформи для впевненості у їхній правильності та ефективності. Це включає тестування

реєстрації користувачів, публікації контенту, коментування, пошуку і т. д.

Сумісність: впевненість у тому, що платформа працює на різних пристроях (комп'ютери, планшети, мобільні телефони) та в різних браузерах без будь-яких проблем. [30]

Тестування безпеки: перевірка на наявність потенційних уразливостей в системі, щоб запобігти можливим атакам або витокам даних.

Тестування швидкодії: Визначення швидкості завантаження сторінок та реагування платформи на дії користувачів для забезпечення високої продуктивності. [12]

Відгуки користувачів: збір фідбеку від реальних користувачів під час тестування для виявлення їхніх вражень, пропозицій та проблем, які вони можуть зустріти.

Після завершення тестування важливо проаналізувати результати та внести необхідні зміни та вдосконалення перед запуском платформи. Такий цикл тестування та вдосконалення може бути повторений декілька разів досягнення оптимального рівня якості та функціональності продукту.

2.2 Модель та концепція функціонування онлайн-платформи

Давайте розглянемо концептуальну модель роботи застосунку. Є клієнт та JSON сервер який займається як зберіганням та і обробкою запитів (рис. 2.1). [31]

Далі розглянемо модель участі користувачів. Модель участі користувачів моєї платформи розрахована на активну взаємодію та співпрацю між учасниками. Кожен користувач може брати участь на різних рівнях залучення, що дозволяє кожному знайти оптимальний для себе формат участі.

Створення та публікація контенту: Користувачі можуть створювати та публікувати свої власні статті, блоги, відеоуроки, кодові приклади або інші матеріали, пов'язані з інформаційно-технологічною галуззю.

Коментування та обговорення: кожен користувач може залишати коментарі, виражати свою думку щодо опублікованого контенту та приймати

участь в обговореннях з іншими учасниками спільноти.

Оцінювання та рецензування: учасники мають можливість оцінювати контент, залишати відгуки та рецензувати матеріали, що допомагає покращувати якість інформації та визначати найбільш цікавий та корисний контент.



Рисунок 2.1 – Концептуальна модель

Поділ досвіду та знань: користувачі можуть ділитися своїм досвідом, навичками та знаннями з іншими учасниками платформи через написання рецензій, створення уроків, відповіді на запитання тощо. [13]

Тепер давайте детально поговоримо про механізми стимулювання участі:

- Система нагород та бонусів: користувачі можуть отримувати бонуси або нагороди за активність на платформі, таку як публікація важливого контенту, допомога іншим користувачам, або активність в обговореннях. Це може включати бали, зібрані за допомогою відгуків та взаємодії з іншими учасниками. [43]

- Статус експерта: платформа може надавати користувачам, які зробили

значний внесок у спільноту або проявили експертні знання, статус експерта. Це може дати їм додатковий авторитет серед інших користувачів та підвищити їхню видимість.

- Конкурси та акції: платформа може організовувати регулярні конкурси, акції або виклики для користувачів. Це може стимулювати активність та творчість, а також привертати нових учасників до спільноти.

- Підтримка інтерактивних функцій: додавання інтерактивних можливостей, таких як голосування, опитування або інтерактивні вебінари, може стимулювати більш активну участь користувачів та сприяти залученню нових учасників.

- Загальні цілі та винагороди: задавання загальних цілей для спільноти, таких як досягнення певної кількості публікацій або коментарів, може стимулювати спільноту до спільної діяльності. Потім можна надати винагороди або відзнаки тим користувачам, які досягли цих цілей.

Давайте поговоримо про можливі функції модерації та контролю якості.

- Перевірка вмісту перед публікацією: усі нові статті, коментарі та інші матеріали можуть проходити перевірку модераторами перед їх публікацією на платформі. Це допомагає уникнути поширення спаму, некоректного контенту та іншого небажаного матеріалу. [44]

- Система відгуків та скарг: користувачі можуть подавати відгуки або скарги щодо контенту або поведінки інших користувачів. Модератори відстежують ці скарги та вживають відповідних заходів для вирішення проблем.

- Встановлення правил спілкування: платформа може мати чіткі правила щодо спілкування користувачів, які визначають допустимі форми взаємодії та забороняють образливе чи некоректне поведінку. [14]

- Автоматизовані системи виявлення спаму та шкідливого контенту: використання програмних засобів для виявлення спаму, вірусів, фішингу та іншого шкідливого контенту може допомогти модераторам швидше виявляти та видаляти такий вміст.

- Комуніті-модерація: залучення досвідчених користувачів до процесу модерації, які можуть допомагати відстежувати та видаляти некоректний контент або порушення правил спілкування.

Наступною властивою складовою є залучення експертів та лідерів галуз. Залучення експертів та лідерів галузі на онлайн-платформу - це ключовий крок у підвищенні авторитету та привабливості спільноти. Ось кілька способів, якими це можна зробити:

1. Запрошення експертів на виступи. Платформа може активно запрошувати відомих фахівців та лідерів галузі для виступів на вебінарах, конференціях або віртуальних зустрічах. Це надасть користувачам можливість отримати цінні знання та поради від досвідчених професіоналів.

2. Гостьові публікації та інтерв'ю. Платформа може пропонувати відомим експертам можливість розміщення гостьових статей, блогів або інтерв'ю, де вони можуть поділитися своїми знаннями та досвідом з аудиторією.

3. Створення експертних панелей. Платформа може формувати експертні панелі з провідних фахівців галузі для відповідей на питання користувачів або проведення дискусій з актуальних тем.

4. Нагородження статусом експерта. Платформа може встановити програму нагородження статусом експерта для активних та впливових учасників, які активно допомагають іншим користувачам та вносять цінний внесок у розвиток спільноти.

Концепція розвитку платформи для обміну досвідом IT-спеціалістів може базуватися на різних стратегічних напрямках. Давайте поговоримо про них. [32]

Розширення функціоналу та зручностей користувачів. Постійне оновлення та розвиток інтерфейсу та функціоналу платформи для забезпечення зручності та ефективності користування. Це може включати вдосконалення системи пошуку, розширення можливостей взаємодії між користувачами, та створення персоналізованих рекомендацій.

Створення сприятливого середовища для експертів та активних учасників. Розвиток програми стимулювання активної участі експертів та лідерів галузі, таких як нагороди за внесок у спільноту, статуси експертів, або можливості для особистого брендування. [33]

Підтримка співпраці та обміну досвідом. Підтримка ініціатив користувачів для спільного вирішення проблем та обміну знаннями та досвідом. Це може включати організацію вебінарів, створення форумів для обговорення актуальних питань, а також можливість співпраці над проектами.

Розширення співпраці зі сторонніми партнерами. Розвиток партнерських відносин з іншими організаціями, комерційними підприємствами та освітніми установами для розширення мережі та обміну ресурсами та експертизою.

2.3 Вибір інструментів для розробки онлайн-платформи

Вибір інструментів для розробки онлайн-платформи є одним із найважливіших кроків у процесі створення успішного продукту. Від правильного вибору залежить не тільки ефективність розробки, але й якість, швидкість та майбутнє супроводження проекту. Різні інструменти мають свої переваги та недоліки, тому ретельний аналіз та порівняння варіантів є важливим етапом.

У наш час три з найбільш популярних інструментів для розробки веб-додатків - це React, Vue та Angular. Кожен з них має свої унікальні особливості та призначення, і вибір між ними залежить від конкретних потреб вашого проекту, ваших технічних уподобань та досвіду вашої команди розробників. [45]

React - це JavaScript-бібліотека для створення інтерфейсів користувача, яка розроблена компанією Facebook. Вона використовується для побудови односторінкових додатків (SPA) та інтерфейсів з високою динамікою. Основні переваги React включають:

Flexibility (Гнучкість): React не є повноцінним фреймворком, тому він дозволяє розробникам використовувати його разом з іншими бібліотеками та

інструментами за їхніми потребами. [15]

Virtual DOM (Віртуальний DOM): Використання віртуального DOM дозволяє React оновлювати тільки ті частини сторінки, які змінилися, що забезпечує високу продуктивність додатків. [38]

Active Community (Активна спільнота): React має велику та активну спільноту розробників, що забезпечує швидку підтримку, багатоцільовий підхід та доступ до багатьох сторонніх бібліотек та рішень.]

До недоліків можна віднести:

- Високий поріг входження: Використання React може вимагати додаткового часу та зусиль для освоєння концепцій, таких як JSX та віртуальний DOM.

- Необхідність вибору додаткових бібліотек: React сам по собі є бібліотекою і не має вбудованих рішень для таких завдань, як маршрутизація чи керування станом, що може вимагати вибору та інтеграції додаткових бібліотек. [47]

Vue - це прогресивний JavaScript-фреймворк для створення користувацьких інтерфейсів. Він позиціонується як простий та легкий у використанні інструмент для розробки SPA та інших веб-додатків. [16] Основні переваги Vue включають:

- Ease of Learning (Легкість навчання): Vue має простий та зрозумілий API, що робить його ідеальним для початківців або розробників, які шукають швидкий старт.

- Versatility (Універсальність): Vue може використовуватися для розробки як великих, так і малих додатків, а також вбудовуватися в існуючі проекти, що робить його універсальним інструментом для різних випадків використання.

- Performance Optimization (Оптимізація продуктивності): Vue має вбудовані механізми для оптимізації продуктивності, такі як кешування компонентів та ліниве завантаження, що дозволяє створювати ефективні та швидкодіючі додатки.

Серед недоліків даного фреймворку хотілось би зазначити:

- Менша екосистема: Хоча Vue має активну та швидко зростаючу спільноту, вона все ще менша, ніж у React або Angular, що може вплинути на доступність сторонніх бібліотек та ресурсів.

- Неявність стандарту: У Vue може відсутність такого ж стандарту, як у React або Angular, що може призвести до різноманітності в підходах розробки між різними проектами. [35]

Angular - це JavaScript-фреймворк для створення веб-додатків, розроблений компанією Google. Angular надає великий набір інструментів та можливостей для розробки великих та складних додатків. [17] Основні переваги Angular включають:

- Full-fledged Framework (Повноцінний фреймворк): Angular надає все необхідне для створення веб-додатків, включаючи маршрутизацію, валідацію форм, кешування HTTP-запитів тощо.

- Two-way Data Binding (Двостороннє зв'язування даних): Angular використовує двостороннє зв'язування даних, що дозволяє автоматично оновлювати дані на сторінці при їх зміні.

- TypeScript Support (Підтримка TypeScript): Angular розроблений з використанням TypeScript, що дозволяє розробникам писати більш безпечний та структурований код. [37]

Його недоліки:

- Складність та обмеженість: Angular є повноцінним фреймворком і може вважатися важким для вивчення та використання. Також, його повнота може бути зайвою для менших проектів або тих, де потрібна більша гнучкість.

- Високий поріг вхідного бар'єра: Використання Angular вимагає знань TypeScript, який є строго типізованою мовою програмування, що може виявитися важким для новачків або команд, які не мають досвіду з TypeScript.

Для розробки платформи для обміну досвідом IT спеціалістів я обрав саме фреймворк React. Мій вибір обумовлений кількома ключовими аспектами, які впливають на успішність та ефективність мого проекту.

По-перше, React відомий своєю гнучкістю та простотою використання. Це дозволяє розробникам швидко створювати складні веб-додатки без великого обсягу коду. Велика кількість готових компонентів та модулів дозволяє значно спростити розробку та покращити продуктивність.

Друге, широка екосистема React, включаючи різноманітні бібліотеки, інструменти та фреймворки, робить його привабливим вибором для розробників. Велика кількість ресурсів та активна спільнота розробників дозволяють швидко знаходити рішення для будь-яких проблем і питань, що виникають під час розробки. [46]

Третє, використання віртуального DOM у React дозволяє забезпечити високу ефективність та швидкість роботи додатку. Відсутність необхідності повного перерендерингу сторінки при зміні даних дозволяє покращити продуктивність та реагування додатку на користувацькі взаємодії.

РОЗДІЛ 3.

РОЗРОБКА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ОНЛАЙН-ПЛАТФОРМИ ОБМІНУ ДОСВІДУ ІТ-СПЕЦІАЛІСТІВ

3.1. Дизайнерські рішення проєкту онлайн-платформи

Так як я не дизайнер, а front-end розробник, тому я буду спиратись на дизайни, які вже існують. За основу я візьму дизайн систему компанії Apple, яка називається “Human Interface Guidelines”. [36] Також я буду надихатись такими застосунками з екосистеми Apple як iCloud та Apple Music.

Давайте детально розглянемо основні елементи інтерфейсу. В мене буде бокова панель для перемикання між секціями сайту (рис 3.1).

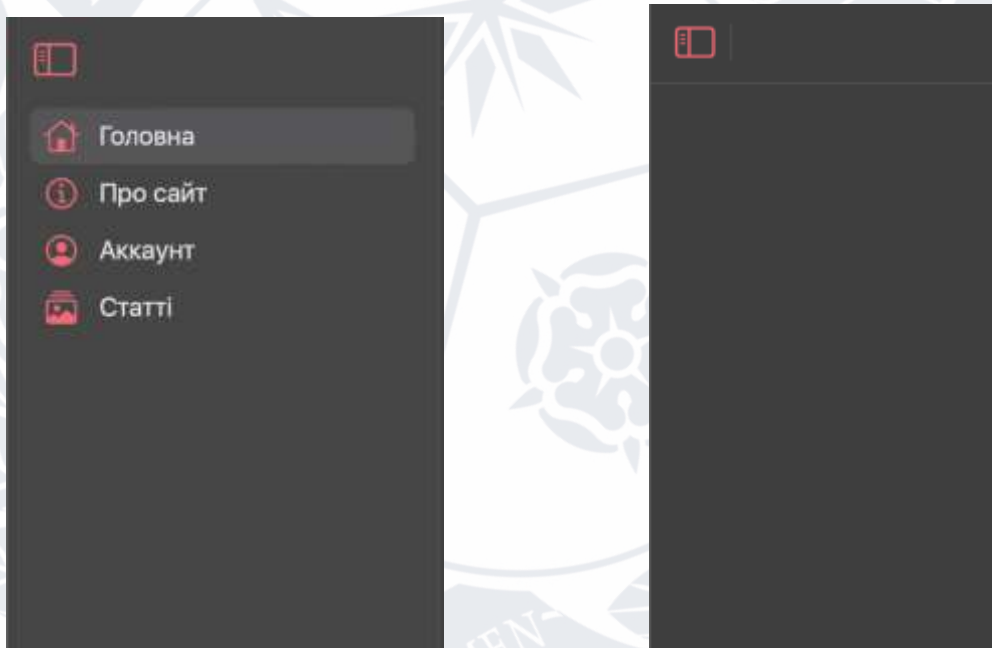


Рисунок 3.1 – Бокова панель в розгорнутому та згорнутому вигляді

В шапці з права будуть кнопки для відкриття меню налаштувань, профілю та нагадувань (рис 3.2 - 3.4).

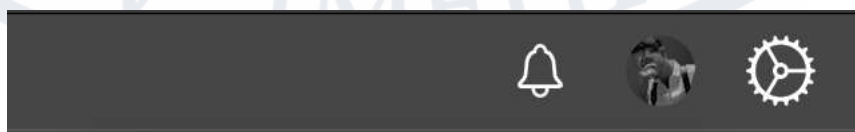


Рисунок 3.2 – Кнопки для відкриття відповідних меню

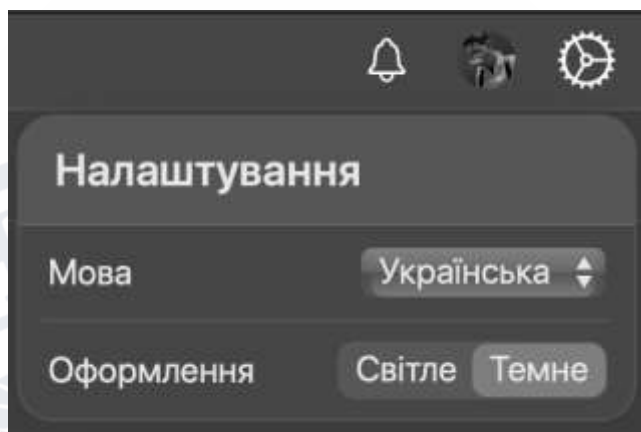


Рисунок 3.3 – Налаштування

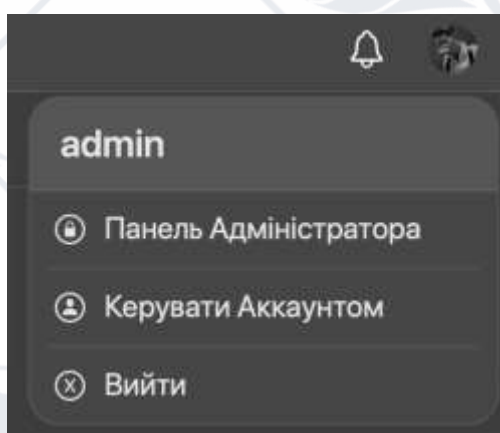


Рисунок 3.4 – Меню акаунту

Далі розглянемо елементи форм та керування (рис 3.5 – 3.8).



Рисунок 3.5 – Вигляд кнопок

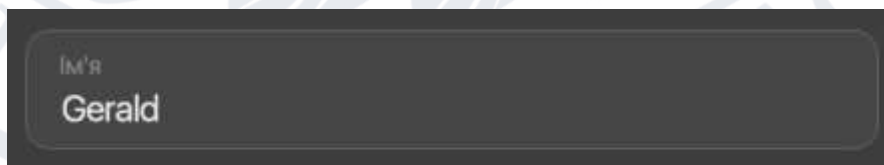


Рисунок 3.6 – Вигляд неактивних полів для вводу даних

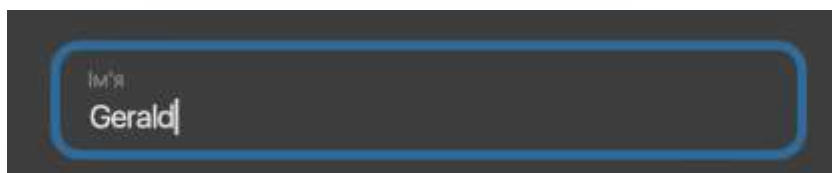


Рисунок 3.7 – Вигляд активних полів для вводу даних

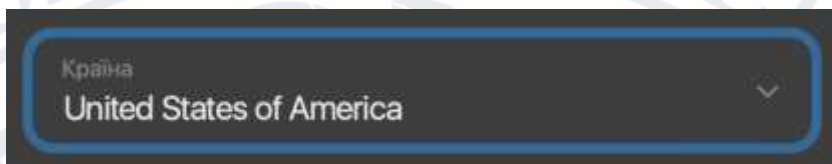


Рисунок 3.8 – Вигляд селекта для вибору значення зі списку

Розглянемо списки для вибору сортування та категорії і перемикач відображення (рис 3.9 – 3.11).

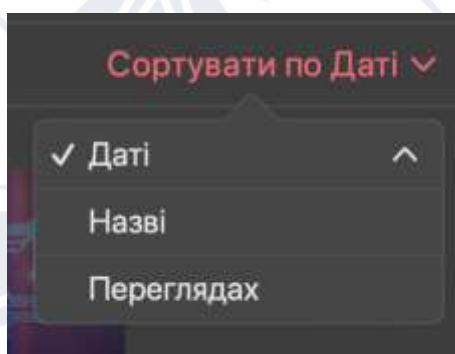


Рисунок 3.9 – Вибір сортування

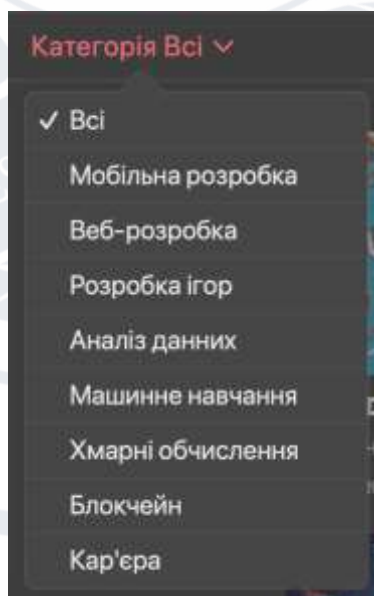


Рисунок 3.10 – Вибір категорії



Рисунок 3.11 – Перемикач сортування

Далі розглянемо дизайн пошуку по статтям (рис 3.12).



Рисунок 3.12 – Пошук по статтям

І також розглянемо один з найголовніших елементів це карточка статті в компактному форматі та розгорнутому (рис. 3.13 – 3.14).

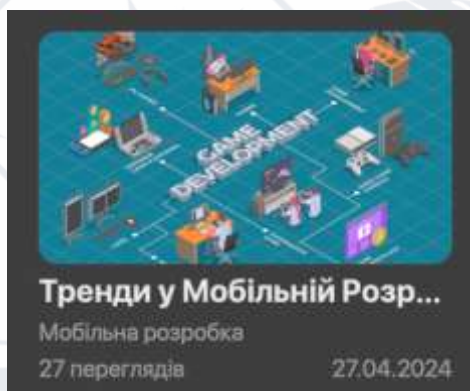


Рисунок 3.13 – Карточка в компактному форматі



Рисунок 3.14 – Карточка в розгорнутому форматі

В компоненті статті присутні звичайні текст та зображення, а також рейтинг та коментар (рис. 3.15 – 3.16).



Рисунок 3.15 – Компонента рейтингу

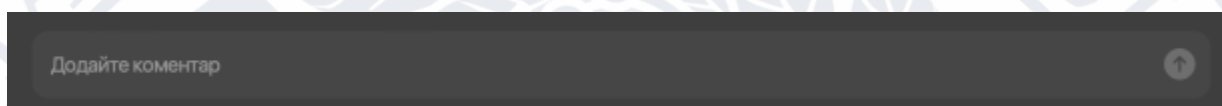


Рисунок 3.16 – Поле для коментаря

Також давайте поговоримо про адаптивність. При зменшенні ширини екрану, деякі компоненти змінюють свій вигляд або взагалі зникають з сторінки.

Для прикладу навігаційна панель на планшетах пристроях відображається внизу а на мобільних телефонах в вигляді списку з вибором (рис 3.17 – 3.18).

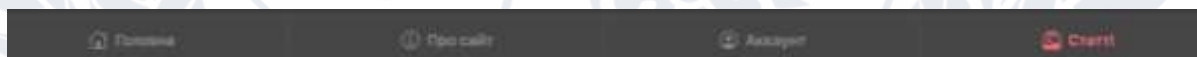


Рисунок 3.17 – Меню на планшетах

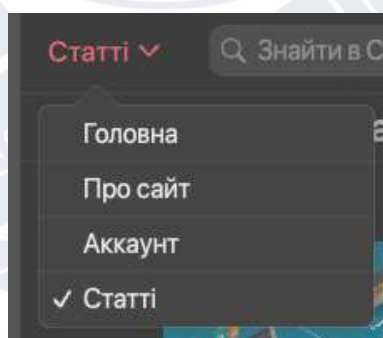


Рисунок 3.18 – Меню на телефоні

Для зручності використання, списки меню та акаунта мають наступний вигляд на телефоні (рис. 3.19).

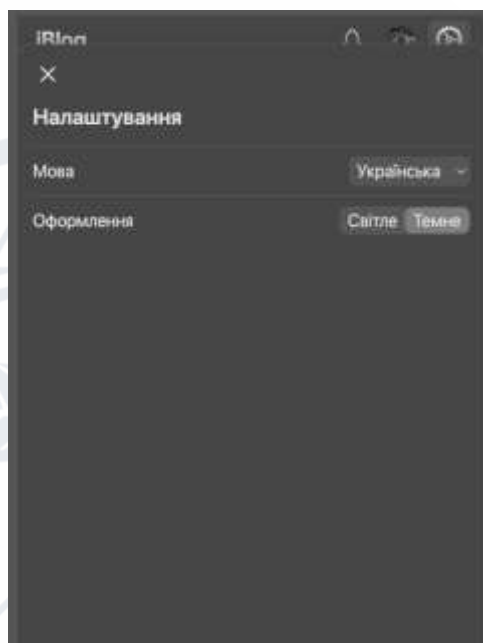


Рисунок 3.19 – Налаштування на телефоні

Остання компонента – це так званий лоадер, коли йде запит на отримання даних то я його відображаю щоб користувач розумів що йде завантаження (рис. 3.20).

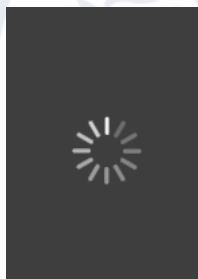


Рисунок 3.20 – Лоадер

3.2 Реалізація серверної частини онлайн-платформи

Для серверної частини додатку я вирішив обрати JSON Server. Це інструмент, який дозволяє швидко створювати фейкові REST API на основі JSON файлів.

Розглянемо детально код сервера (рис 3.21).

```
const fs = require('fs');
const jsonServer = require('json-server');
const path = require('path');

const server = jsonServer.create();

const router = jsonServer.router(path.resolve(__dirname, 'db.json'));

const PORT = process.env.PORT || 8000;

server.use(jsonServer.defaults({}));
```

Рисунок 3.21 – Код сервера

Тут створюється фейковий сервер REST API з використанням JSON Server. Він використовує файли JSON для зберігання даних. Кожен HTTP запит обробляється цим сервером, використовуючи дані з файлу db.json. Код також встановлює налаштування сервера, такі як маршрутизація та обробка запитів (рис.3.22).

```
server.use(router);
server.listen(PORT, () => {
  console.log('server is running on 8000 port');
});
```

Рисунок 3.22 – Налаштування сервера

Цей фрагмент коду налаштовує сервер, щоб він використовував маршрутизатор для обробки запитів, і потім запускає сервер на певному порту (зазвичай 8000), слухаючи цей порт для вхідних з'єднань (рис 3.23).

```
server.use((req, res, next) => {
  if (!req.headers.authorization) {
    return res.status(403).json({ message: 'AUTH ERROR' });
  }

  next();
});
```

Рисунок 3.23 – Налаштування серверу для обробки запитів

Даний код перевіряє чи в кожному запиті присутній заголовок який показує чи авторизований користувач чи ні.

Також присутній додатковий шлях для входу /login, в який передаються пошта та пароль і якщо даний користувач знаходиться в файлі json то сервер повертає данні а якщо то по повертає помилку.

3.3 Реалізація клієнтської частини онлайн-платформи

Даний проект написаний на React Js, для сховища даних використовується Redux Toolkit, а для написання стилів SCSS. Так як файлів та коду дуже багато, то я опишу головні компоненти та аспекти даного проекту.

Розглянемо структуру сайту.

Мною було створено наступні сторінки:

1. MainPage – головна сторінка
2. AboutPage – про проект
3. AccountPage – налаштування акаунту користувача
4. ArticlesPage – сторінка з всіма статтями
5. ArticleDetailsPage – сторінка одної статті
6. ArticleCreatePage – створення статті

Почнемо з головної сторінки та сторінки про проект. Вони інформаційні, на головній данні для входу в акаунти а на сторінці про сайт посилання на github репозиторій. Також в сховищі даних redux яке я назвав стан застосунку при завантажині сторінки змінюю змінну яка відповідає за показ прелоадера (рис 3.24).

Тут використовується функція useDispatch() для отримання доступу до диспетчера Redux. Диспетчер дозволяє нам відправляти дії (actions) до Redux store для зміни стану додатку.

```

const MainPage = () => {
  const dispatch = useAppDispatch();
  useEffect(() => {
    dispatch(appStateActions.setContentLoaded(true));
  }, [dispatch]);

  return (
    <div className={cls.MainPage}>
      <MainPageHeader />
      <div className={cls.MainPageInfo}>
        <div>
          <h3>Accounts:</h3>
          <br />
          <p>username: user</p>
          <p>password: 123</p>
          <br />
          <p>username: admin</p>
          <p>password: 123</p>
        </div>
      </div>
    </div>
  );
}

```

Рисунок 3.24 – Код головної сторінки

В AboutPage код схожий тільки компонента повертає другий html (рис 3.25).

```

const AboutPage = () => {
  ...

  return (
    <div className={cls.AboutPage}>
      <AboutPageHeader />
      <div className={cls.AboutPageInfo}>
        <div>
          <a
            href="https://github.com/kornienkokostia/production-project"
            target="_blank"
            rel="noreferrer"
          >
            Git Hub Repo
          </a>
        </div>
      </div>
    </div>
  );
}

```

Рисунок 3.25 – Код сторінки з інформацією про сайт

Розглянемо сторінку профіля користувача. Дана компонента приймає `id` користувача як параметр. Далі проходить отримання даних про користувача, а також змінної чи завантажились данні і чи виникла помилка (рис. 3.26)

```
export const EditableAccountCard = memo((props: AditableAccountCardProps) => {
  const { id } = props;
  const dispatch = useAppDispatch();

  useEffect(() => {
    if (id) {
      dispatch(fetchAccountData(id));
    }
  }, [dispatch, id]);

  const formData = useSelector(getAccountForm);
  const isLoading = useSelector(getAccountIsLoading);
  const error = useSelector(getAccountError);
  ...
});
```

Рисунок 3.26 – Код сторінки профілю

Далі створюємо функції для збереження нових даних коли акаунт редагують (рис. 3.27).

```
const onChangeFirstName = useCallback(
  (val?: string) => {
    dispatch(accountActions.updateAccount({ firstname: val || "" }));
  },
  [dispatch],
);
```

Рисунок 3.27 – Код зміни поля для вводу профілю

`useCallback` - це хук React, який повертає мемоізовану версію зазначеної функції. Його основна мета полягає в уникненні зміни референції на функцію під час повторних рендерів, коли це необхідно.

Коли використовуємо `useCallback`, ми повертаємо мемоізовану версію функції, яка буде залишатися такою самою між рендерами, якщо масив залежностей не зміниться.

І далі передаємо ці функції та данні в компоненту яка відмалює карточку користувача (рис. 3.28).

```

return (
  <DynamicModuleLoader reducers={initialReducers}>
    <AccountCard
      data={formData}
      error={error}
      onChangeFirstName={onChangeFirstName}
      onChangeLastName={onChangeLastName}
      onChangeAge={onChangeAge}
      onChangeCity={onChangeCity}
      onChangePhoto={onChangePhoto}
      onChangeUsername={onChangeUsername}
      onChangeCurrency={onChangeCurrency}
      onChangeCountry={onChangeCountry}
      readonly={readonly}
      formErrors={formErrors}
    />
  </DynamicModuleLoader>
);

```

Рисунок 3.28 – Код виклику компоненти карточки користувача

Розглянемо сторінку редагування акаунту в браузері (рис. 3.29).

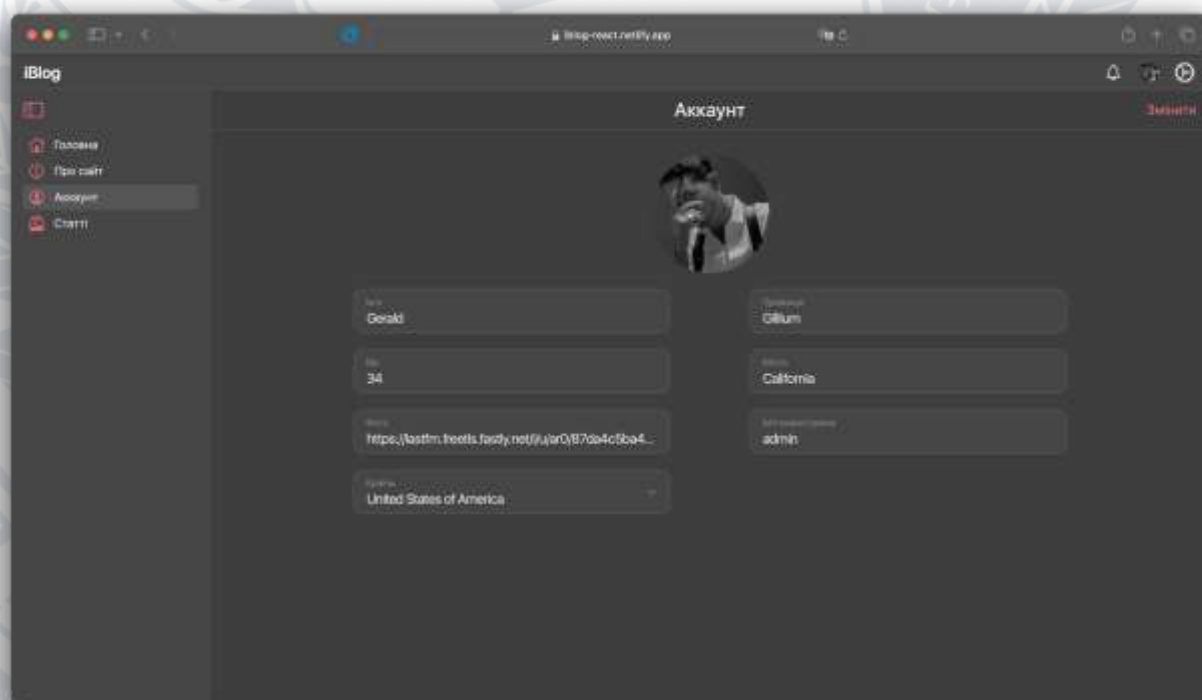


Рисунок 3.29 – Вид сторінки редагування акаунту

Наступною розглянемо сторінку з статтями. В даній компоненті присутні 2 елементи це фільтри (<ArticlesPageFilters />)

І також список всіх статей з пагінацією (<ArticlesInfiniteList />)

В компоненті з фільтрацією є наступне:

- Перемикач виду статей чи компактний чи розгорнутий (<ArticleViewSwitcher />) в який ми передаємо змінну виду яка зберігається в сторі та функцію для зміни значення яка також задає значення поточній сторінці – 1 (рис. 3.30).

```
export const ArticlesPageFilters = memo((props: ArticlesPageFiltersProps) => {
  ...
  const view = useSelector(getArticlesPageView);
  const onChangeView = useCallback(
    (view: ArticleView) => {
      dispatch(articlesPageActions.setView(view));
      dispatch(articlesPageActions.setPage(1));
      fetchData();
    },
    [dispatch, fetchData],
  );
  ...
  return (
    ...
    <ArticleViewSwitcher
      view={view}
      onViewClick={onChangeView}
      className={cls.ArticleViewSwitcher}
    />
  );
});
```

Рисунок 3.30 – Перемикач режиму відображення

- Пошук який складається з іконки пошуку, поля для вводу і кнопки для рчищення яка показується коли поле не пуста (рис. 3.31).

```

<div className={cls.search}>
  <SearchIcon className={cls.searchIcon} />
  <input
    type="text"
    placeholder={t('Find in Articles')}
    className={cls.searchInput}
    value={search}
    onChange={onChangeSearch}
  />
  <Button
    theme="clear"
    className={cls.clearBtn}
    disabled={!search.length > 0}
    onClick={onClearSearch}>
    <ClearInputIcon className={cls.clearBtnIcon} />
  </Button>
</div>

```

Рисунок 3.31 – Пошук

- Вибір категорії статей в який ми передаємо по аналогії з перемикачем вигляду значення категорії поточне та функцію для зміни (рис. 3.32).

```

const type = useSelector(getArticlesPageType);
...
const onChangeCategory = useCallback(
  (type: ArticleType) => {
    dispatch(articlesPageActions.setType(type));
    dispatch(articlesPageActions.setPage(1));
    fetchData();
  },
  [dispatch, fetchData],
);
...
return (
  ...
  <ArticleCategorySelector
    value={type}
    onChange={onChangeCategory}
    sidebarPadding={!navbarCollapsed}
    withoutAll={false}
  />

```

Рисунок 3.32 – Вибір категорії статей

- Остання компонента це сортування, вибір яких здійснюється по аналогії з вибором категорії (рис. 3.33)

```
const sort = useSelector(getArticlesPageSort);
const order = useSelector(getArticlesPageOrder);
...
const onChangeSort = useCallback(
  (sort: ArticleSortField) => {
    dispatch(articlesPageActions.setSort(sort));
    dispatch(articlesPageActions.setPage(1));
    fetchData();
  },
  [dispatch, fetchData],
);

const onChangeOrder = useCallback(
  (order: SortOrder) => {
    dispatch(articlesPageActions.setOrder(order));
    dispatch(articlesPageActions.setPage(1));
    fetchData();
  },
  [dispatch, fetchData],
);
...
return (
  ...
  <ArticleSortSelector
    value={sort}
    order={order}
    onChange={onChangeSort}
    onChangeOrder={onChangeOrder}
  />
```

Рисунок 3.33 – Сортування

Далі перейдемо до компоненти зі статтями.

Сочатку візьмемо данні зі стору а саме статті, змінну яка показує чи завантажились статті, чи виникла помилка та вид відображення (рис. 3.34).

```
const articles = useSelector(getArticles.selectAll);
const isLoading = useSelector(getArticlesPagelsLoading);
const error = useSelector(getArticlesPageError);
const view = useSelector(getArticlesPageView);
```

Рисунок 3.34 – Головні змінні на сторінці статті

Далі передамо їх як параметри в компоненту для рендерингу їх (<ArticleList />) (рис. 3.35)

```
return (
  <ArticleList
    view={view}
    isLoading={isLoading}
    articles={articles}
    className={className}
  />
);
```

Рисунок 3.35 – Рендер списку статей

В цій компоненті ми відмальовуємо лоадер поки статті завантажуються наступним чином (рис. 3.36).

```
{isLoading && articles.length && (
  <div className={cls.loading}>
    <Loader />
  </div>
)}
```

Рисунок 3.36 – Лоадер в компоненті статей

Якщо данні завантажились то кожне елемент масиву з статтями ми відмальовуємо за допомогою функції renderArticle яка приймає статтю, вид в якому відмальовувати її та ключ, значення якого дорівнює ID статті, для коректного рендерингу списку (3.37).

```
const renderArticle = (article: Article) => (
  <ArticleListItem article={article} view={view} key={article.id} />
);
```

Рисунок 3.37 – Рендер списку статей

І викликаємо цю функцію в циклі (3.38).

```
<div className={classNames(cls.ArticleList, {}, [cls.view])}>
  {articles.length > 0 ? articles.map(renderArticle) : null}
</div>
```

Рисунок 3.38 – Цикл по масиву статей

Давайте ще поговоримо про компоненту `articleListItem`. В залежності від переданого виду відображення є 2 варіанти рендеру:

1. Якщо вид деталізований, то рендеримо аватар користувача, дату створення, назва, зображення, один абзац тексту, кількість переглядів та конопка щоб читати далі. І це все розтягнуте не всю ширину блоку з статтями.

```
return (
  <div
    className={classNames(cls.ArticleListItem, {}, [className, cls.view])}>
    <div className={cls.card}>
      <div className={cls.header}>
        <AccountPhoto
          src={article.user.avatar}
          className={cls.accountPic}
        />
        <p className={cls.username}>{article.user.username}</p>
        <p className={cls.date}>{article.createdAt}</p>
      </div>
      <h3 className={cls.title}>{article.title}</h3>
      {types}
      <img src={article.img} className={cls.image} alt={article.img} />
      {textBlock && (
        <ArticleTextBlockComponent
          block={textBlock}
          className={cls.textBlock}
        />
      )}
      <div className={cls.footer}>
        <AppLink
          to={getRouteArticleDetails(article.id)}
          state={{ prevPath: location.pathname }}
          className={cls.link}
          theme="apple-link">
          <span>{t('Read more')}</span>
          <ArrowLinkIcon className={cls.linkIcon} />
        </AppLink>
      </div>
    </div>
  </div>
);
}
```

Рисунок 3.38 – Код для розгорнутого відображення

А якщо вид стислий, тобто в блок `if` ми не заходимо, то рендеримо від 3

до 1 статей в залежності від розміру екрану в одну стрічку. Відображаємо тільки зображення, назву, категорію, кількість переглядів та дату створення (рис. 3.39).

```
return (
  <div
    className={classNames(cls.ArticleListItem, {}, [className, cls.view])}>
    <AppLink
      className={cls.card}
      to={getRouteArticleDetails(article.id)}
      state={{ prevPath: location.pathname }}>
      <div className={cls.imageWrapper}>
        <img src={article.img} className={cls.image} alt={article.title} />
      </div>
      <h3 className={cls.title}>{article.title}</h3>

      {types}
      <div className={cls.infoWrapper}>
        {views}
        <p className={cls.date}>{`${article.createdAt}`}</p>
      </div>
    </AppLink>
  </div>
);
```

Рисунок 3.39 – Код стислого відображення
Вигляд сторінки з статтями (рис. 3.40)

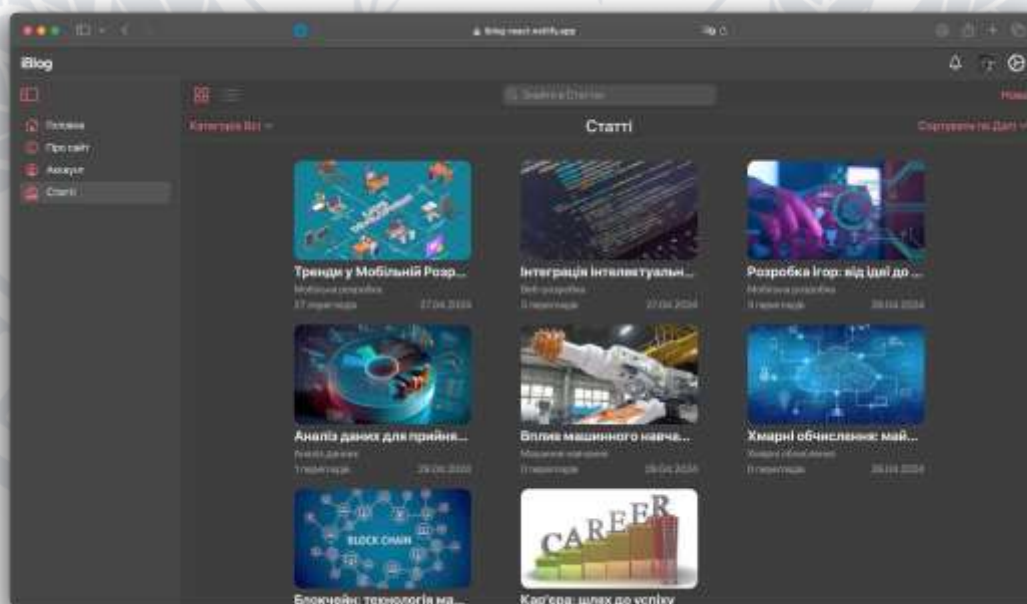


Рисунок 3.40 – Вид сторінки списку статей

Наступною розберемо сторінку одної статті. Дана сторінка виступає контейнером для різних компонент, таких як деталі статті - `ArticleDetails`, список рекомендованих статей для прочитання - `ArticleRecommendationsList`, рейтинг статті - `ArticleRating` і список коментарів - `ArticleDetailsComments` (рис. 3.41).

```
const ArticleDetailsPage = ({ className }: ArticleDetailsPageProps) => {
  return (
    <DynamicModuleLoader reducers={reducers}>
      <div className={classNames(cls.ArticleDetailsPage, {}, [className])}>
        <ArticleDetailsPageHeader />
        <section className={cls.ArticleDetailsWrapper} ref={wrapperRef}>
          <ArticleDetails id={id} />
          <ArticleRecommendationsList />
          <ArticleRating articleId={id} />
          <ArticleDetailsComments id={id} />
        </section>
      </div>
    </DynamicModuleLoader>
  );
};
```

Рисунок 3.41 – Компоненти які містить сторінка статті

Давайте розглянемо кожен блок більш детально.

`ArticleDetails` блок рендерить вміст статті, який може складатись з послідовності 3 видів блоків: блок тексту, блок зображення, та блок коду (рис. 3.42).

```
const renderBlock = useCallback((block: ArticleBlock) => {
  switch (block.type) {
    case ArticleBlockType.CODE:
      return <ArticleCodeBlockComponent key={block.id} block={block} />;
    case ArticleBlockType.IMAGE:
      return <ArticleImageBlockComponent key={block.id} block={block} />;
    case ArticleBlockType.TEXT:
      return <ArticleTextBlockComponent key={block.id} block={block} />;
    default:
      return null;
  }
}, []);
```

Рисунок 3.42 – Відображення блоку статті в залежності від виду

В списку рекомендацій я застосовую ту ж компоненту що і використовувалась для відображення списку статей (рис. 3.43).

```
return (
  <div
    className={classNames(cls.ArticleRecommendationsList, {}, [className])}>
    <h1 className={cls.title}>{t('Recommendations')}</h1>
    <ArticleList
      articles={articles}
      isLoading={isLoading}
      view={ArticleView.SMALL}
      className={cls.recommendations}
    />
  </div>
);
```

Рисунок 3.43 – Відображення рекомендацій

Для відображення рейтингу використаю компоненту RatingCard, яка виводить 5 зірок і в відповідності чи є вже рейтинг від даного користувача заповнює їх кольором (рис. 3.44).

```
return (
  <RatingCard
    rate={rating?.rate}
    title={t('Rate this article')}
    feedbackTitle={t('Leave a feedback')}
    placeholder={t('Enter your feedback')}
    onCancel={onCancel}
    onAccept={onAccept}
    hasFeedback
    className={cls.ArticleRating}
  />
);
```

Рисунок 3.44 – Передача значень в компоненту рейтингу

І остання компонента це відображення CommentList та додавання коментарів AddCommentForm (рис. 3.45).

```
export const ArticleDetailsComments = memo(
  ({ className, id }: ArticleDetailsCommentsProps) => {
    ...
    return (
      <div className={classNames(cls.ArticleDetailsComments, {}, [className])}>
        <h1 className={cls.title}>{t('Comments')}</h1>
        <AddCommentForm onSendComment={onSendComment} />
        <CommentList comments={comments} onSendComment={onSendComment} />
      </div>
    );
  },
);
```

Рисунок 3.45 – Форма для створення коментарів

Блок з вмістом статті наведено на рисунку 3.46.

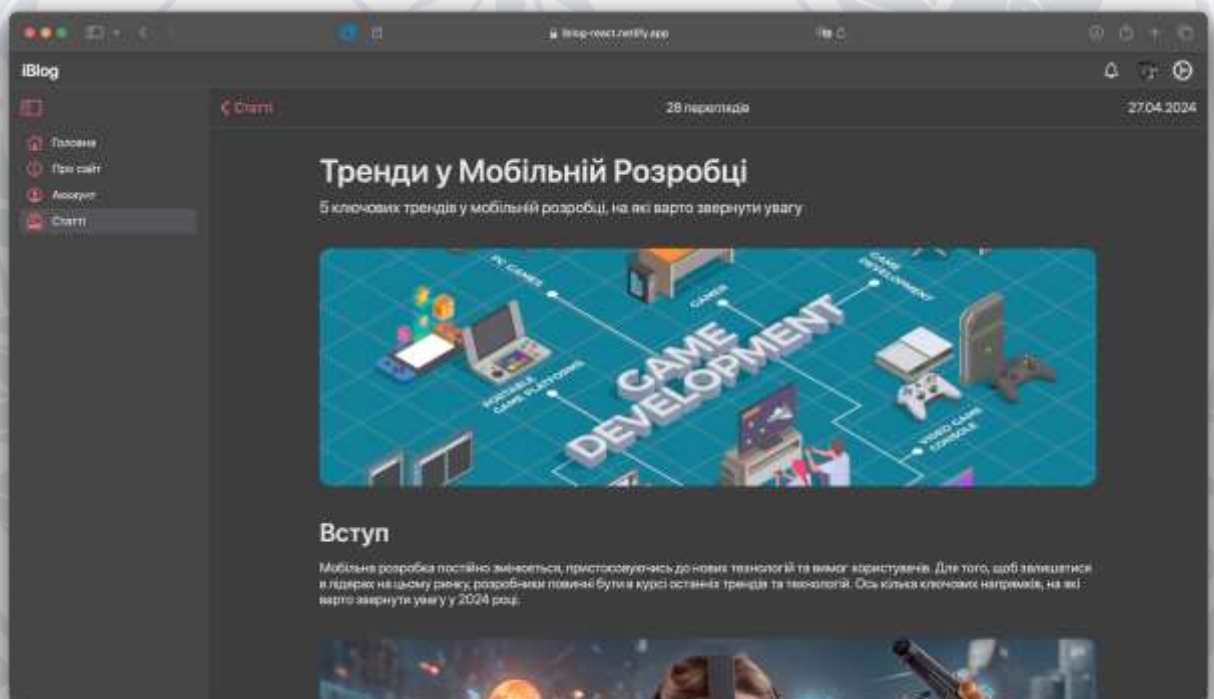


Рисунок 3.46 – Блок з вмістом статті

Блоки з рекомендаціями та коментарями наведено відповідно на рисунках 3.47 та 3.48.

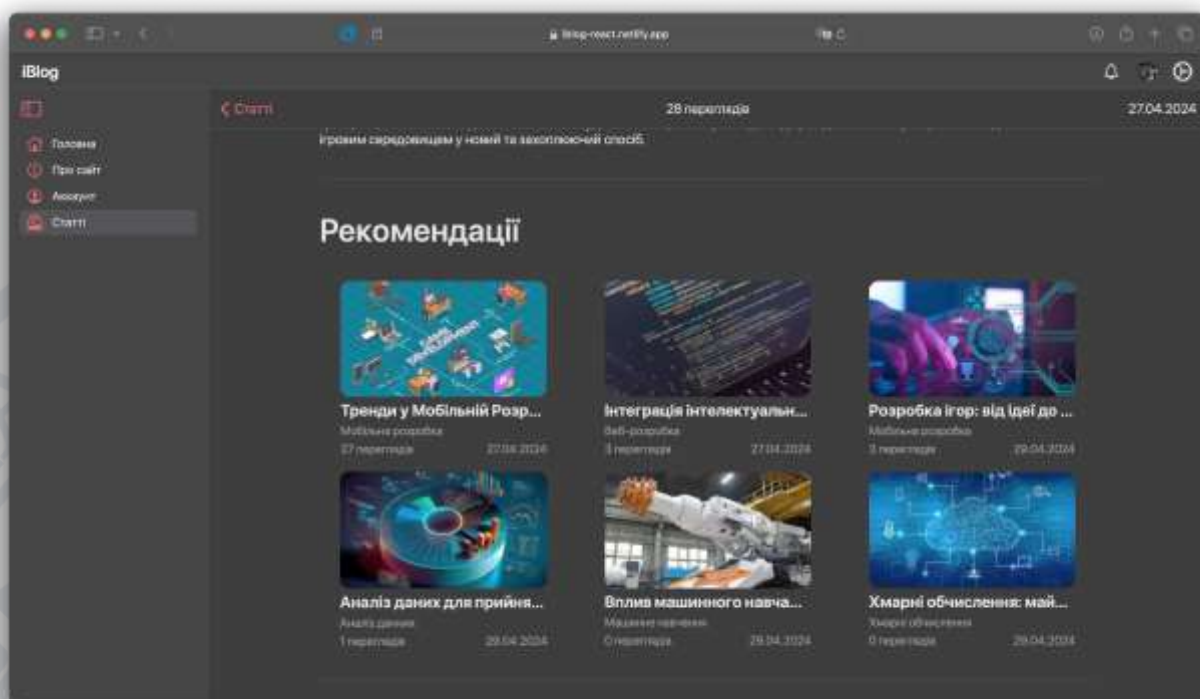


Рисунок 3.47 – Блок з рекомендаціями

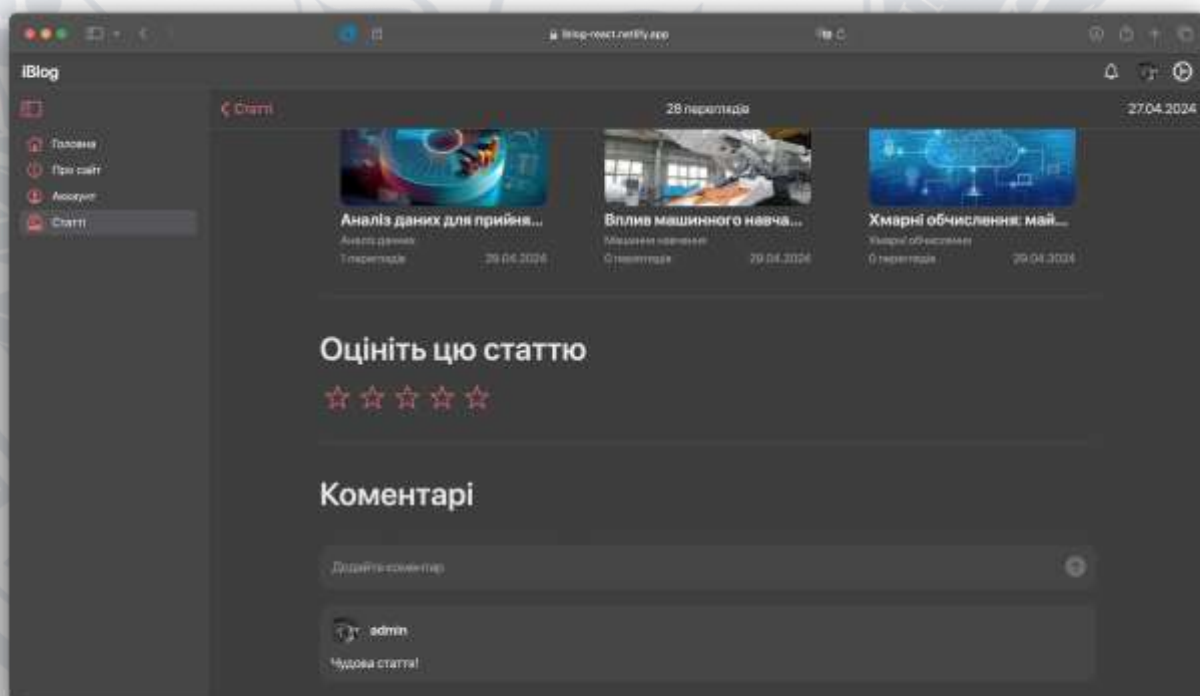


Рисунок 3.48 – Блок з коментарями

Наступний кроком розглянемо сторінку створення статті. Дана сторінка містись в компоненті заголовку селект з вибором категорії а також селект для додавання блоку (рис. 3.49).

```
...
    <ArticleCategorySelector
      value={category[0]}
      onChange={onChangeCategory}
      sidebarPadding={!navbarCollapsed}
      withoutAll={true}
    />
    <ArticleAddBlockSelector
      value={ArticleBlockType.TEXT}
      onChange={onAddBlock}
    />
  ...
```

Рисунок 3.49 – Вибір категорії та додавання блоку статті

І далі йдуть поля для створення статті такі як заголовок, підзаголовок, зображення головне. А після них потрібно обрати тип блоку щоб відповідні компоненти відмалювались. Ось приклад коду з заголовком та під заголовком (рис. 3.50)

```
return (
  <div className={classNames(cls.ArticleCreateEditWrapper, {}, [])}>
    <div className={cls.ArticleCreateEdit}>
      <div className={cls.ArticleEditField}>
        <h3 className={cls.ArticleEditFieldName}>{t('Title')}</h3>
        <TextArea
          placeholder={t('Enter text')}
          value={title}
          onChange={val => dispatch(articleEditActions.setTitle(val))}
        />
      </div>
      <div className={cls.ArticleEditField}>

        <h3 className={cls.ArticleEditFieldName}>{t('Subtitle')}</h3>
        <TextArea
          placeholder={t('Enter text')}
          value={subtitle}
          onChange={val => dispatch(articleEditActions.setSubtitle(val))}
        />
      </div>
    </div>
  </div>
```

Рисунок 3.50 – Поля для створення статті

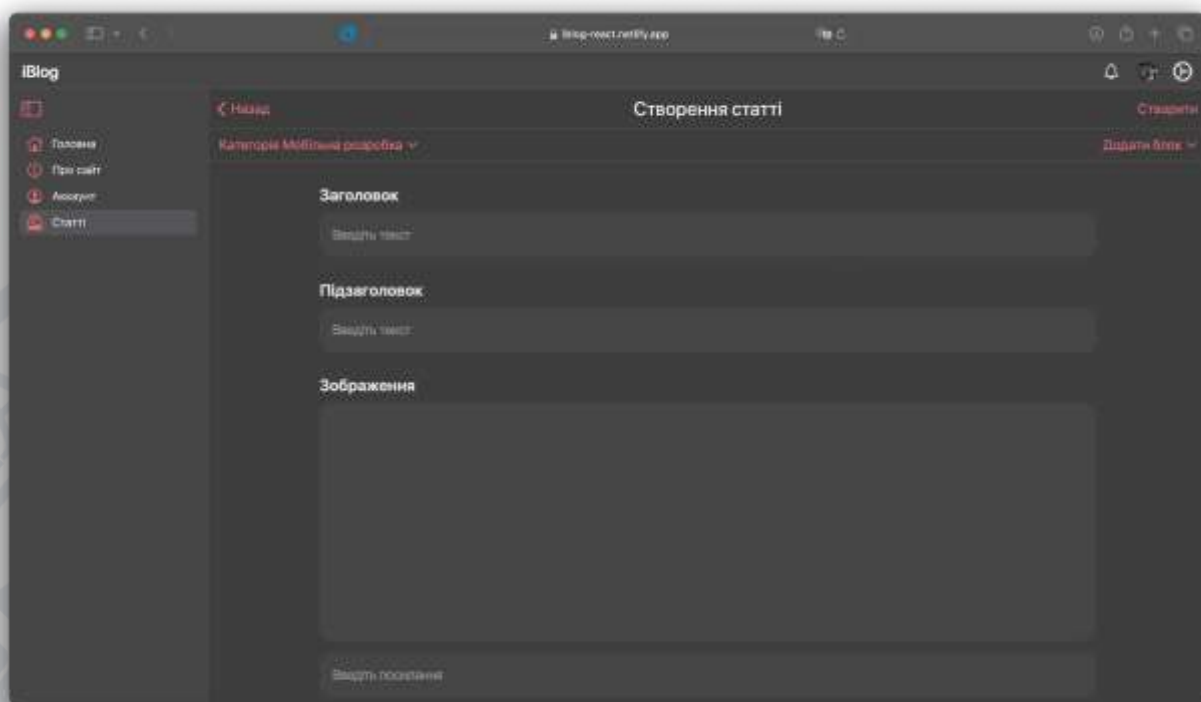


Рисунок 3.51 – Сторінка створення статті

ВИСНОВКИ

У цій бакалаврській роботі досліджено і розроблено онлайн платформу обміну досвідом для IT-спеціалістів. Результати дослідження показують значний потенціал таких платформ у сприянні взаємовідносинам між фахівцями, обміну знаннями та розвитку професійних навичок і можна зробити наступні висновки:

1. В процесі розробки платформи було використано сучасні технології веброботи, такі як React, TypeScript та SCSS. Застосування архітектурного підходу з використанням Redux для керування станом додатку забезпечує надійність та безпеку платформи.

2. Основною функціональністю платформи є можливість обміну досвідом та знаннями через створення профілів користувачів, додавання та перегляд публікацій, коментування та оцінювання матеріалів. Також було реалізовано можливість спілкування між користувачами за допомогою системи приватних повідомлень.

3. Впровадження такої платформи може сприяти зростанню продуктивності та інноваціям в IT-галузі, сприяючи взаємному вивченню та обміну передовими практиками. Однак, для досягнення успіху важливо створити сприятливу спільноту користувачів та розвивати платформу відповідно до їхніх потреб та вимог. Додаткові дослідження та вдосконалення можуть сприяти подальшому розвитку цього інноваційного засобу співпраці та навчання в IT-сфері.

4. У процесі створення платформи були виявлені деякі ключові виклики, які потрібно було вирішити. Одним із них було забезпечення дієвого механізму фільтрації та пошуку контенту, щоб користувачі могли знаходити потрібну інформацію швидко та ефективно. Для цього було розроблено розширену систему категоризації та тегування контенту, яка дозволяє точно визначати тематику та область інтересу.

5. Ще однією важливою аспектом є підтримка співпраці та взаємодії між користувачами. Для цього було впроваджено можливість створення спільнот та груп інтересів, де фахівці можуть обмінюватися ідеями, вирішувати спільні завдання та навіть розробляти проекти разом. Це сприяє збільшенню рівня взаємодії та спільної роботи, що може стимулювати творчий процес та інновації у сфері ІТ.

6. Важливим аспектом є забезпечення безпеки та конфіденційності даних користувачів. Застосування сучасних методів шифрування та механізмів аутентифікації дозволяє зберігати особисту інформацію користувачів в захищеному середовищі, зменшуючи ризик витоку даних та недобросовісного використання інформації. Такий підхід підвищує рівень довіри користувачів до платформи та стимулює активну участь у спільноті.

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Скільки IT-фопів в Україні. URL: <https://dou.ua/lenta/articles/how-many-devs-in-ukraine-2023/>
2. Digital 2023: global overview report. URL: <https://datareportal.com/reports/digital-2023-global-overview-report>
3. These 3 charts show the global growth in online learning. URL: <https://www.weforum.org/agenda/2022/01/online-learning-courses-reskill-skills-gap/>
4. Тенденції розвитку медіа в сучасному світі. URL: https://media-systems.ua/en/news/media_trends_2023
5. E-learning в бізнес-освіті: сучасні форми, тенденції, перспективи. URL: https://ivo.kneu.edu.ua/ua/dosl_glot/temat_zahodi/org2013/
6. У чому секрет успіху Coursera & Udemy і тенденції на ринку онлайн-освіти 2020. URL: <https://platon.ua/news/v-chem-sekret-uspeha-coursera-udemy-i-tendenczii-na-rynke-onlajn-obrazovaniya-2020.html>
7. Огляд популярних інтерактивних платформ. URL: <https://training.qatestlab.com/blog/technical-articles/overview-of-popular-interactive-platforms>
8. Найпопулярніші соцмережі. URL: <https://wedex.com.ua/blog/najpopulyarnishi-sotsmerezhi>
9. Technology platform overview, URL: <https://www.sap.com/uk/products/technology-platform/what-is-a-technology-platform.html#>
10. Розробка онлайн-платформи для освітньої організації «Культурний проект». URL: <https://cases.media/en/case/rozrobka-onlain-platformi-dlya-osvitnoyi-organizaciyi-kulturnii-proekt>
11. How to create a web platform? We collect step-by-step keys for an efficient platform. URL: <https://www.linkedin.com/pulse/how-create-web-platform-we-collect-step-by-step-keys-efficient->

12. Best Practices for Mobile Application Development. URL: <https://www.revelo.com/blog/best-practices-for-mobile-application-development>
13. Platform Economy: The 4 Key Business Models. URL: <https://medium.com/euro-freelancers/platform-economy-the-4-key-business-models-1fc0eda7241e>
14. Online Platforms: Changing the Landscape of the Media Industry. URL: <https://www.everand.com/article/572435128/How-Are-Ott-Platforms-Changing-The-Landscape-Of-The-Media-Entertainment-Industry>
15. Що таке React JS? Як почати вивчати Реакт? Навички для react developer. URL: <https://cases.media/en/article/sho-take-react-js-yak-pochati-vivchati-reakt-navichki-dlya-react-developer>
16. Vue.js як JavaScript-фреймворк. URL: <https://foxminded.ua/vue-js/>
17. Перші кроки в Angular: усе, що потрібно знати розробнику-початківцю. URL: <https://foxminded.ua/angular/>
18. The Top Social Media Trends for 2024 That You Need To Pay Attention To. URL: <https://www.sendible.com/insights/latest-social-media-trends-report>
19. The Top 10 Online Learning Platforms for 2024. URL: <https://www.thinkific.com/blog/online-learning-platforms/>
20. Online Learning Platforms: The Different Types And Their Benefits. URL: <https://www.forbes.com/advisor/education/career-resources/online-learning-platforms/>
21. Online Platforms – Concurrences. URL: <https://www.concurrences.com/en/dictionary/online-platforms>
22. Digital platform | Definition from TechTarget. URL: <https://www.techtarget.com/searchcio/definition/digital-platform>
23. Digital marketplaces: Examples, benefits, strategiess. URL: <https://www.the-future-of-commerce.com/2021/05/12/digital-marketplaces/>
24. Top 11 Online Learning Platforms for Individuals & Businesses. URL: <https://www.learnworlds.com/online-learning-platforms/>

25. Technology Platforms & SaaS. URL: <https://technologyholdings.com/focus-sectors/technology-platforms-and-saas/>
26. Stack Overflow: What is it? How does it work? URL: <https://datascientest.com/en/stack-overflow-what-is-it-how-does-it-work>
27. Should you use Medium or dev.to? URL: <https://dev.to/davidmm1707/should-you-use-medium-or-dev-to-2g1d>
28. Web development methodologies and approaches. URL: <https://www.adcisolutions.com/knowledge/web-development-methodologies-and-approaches>
29. Digital Platform Development Lifecycle. URL: <https://www.efpf.org/post/digital-platform-development-lifecycle>
30. How to Develop an Effective Digital Platform Strategy for Your Business? URL: <https://medium.com/@mudita.seo15/how-to-develop-an-effective-digital-platform-strategy-for-your-business-546b1f7c50ac>
31. How to Use JSON Server for Front-end Development. URL: <https://www.freecodecamp.org/news/json-server-for-frontend-development/>
32. What are Online platforms? URL: <https://corpocrat.com/2022/08/21/what-are-online-platforms/>
33. 4 Ways To Use Online Content Platforms To Bring Value To Your Audience. URL: <https://www.forbes.com/sites/johnhall/2021/07/30/4-ways-to-use-online-content-platforms-to-bring-value-to-your-audience/?sh=540807032159>
34. React The library for web and native user interfaces. URL: <https://react.dev>
35. What Is Vue JS? URL: <https://builtin.com/software-engineering-perspectives/vue-js>
36. Human Interface Guidelines. URL: <https://developer.apple.com/design/human-interface-guidelines>
37. What Is TypeScript? URL: <https://thenewstack.io/what-is-typescript/>
38. ReactJS Virtual DOM. URL: <https://www.geeksforgeeks.org/reactjs-virtual-dom/>
39. What Is Social Networking? URL: <https://www.investopedia.com/terms/s/social-networking.asp>

40. What is a Platform Ecosystem? URL: <https://www.partnerfleet.io/blog/what-is-a-platform-ecosystem>.
41. Плахотніков А. Розробка веб-додатків на основі сучасних технологій. Київ: Видавництво "Факт", 2019. 320 с.
42. Литвиненко О. Веб-програмування для початківців. Львів: Видавництво "Самміт-Книга", 2020. 280 с.
43. Коваленко І. Інтернет-платформи: теорія та практика створення. Київ: Видавництво "Будинок книги", 2019. 400 с.
44. Мельник В. Сучасні веб-технології. Київ: Діафра-К, 2020. 350 с.
45. Петренко О. Онлайн-спільноти: створення та управління. Львів: Видавництво "Світ", 2019. 290 с.
46. Іваненко Ю. Архітектура веб-додатків. Київ: Видавництво "Факт", 2021. 370 с.
47. Бондар С. Модернізація веб-сервісів для обміну інформацією. Львів: Видавництво "Самміт-Книга", 2020. 340 с.
48. Шафорост В. В., Корнієнко К. К., Ніколюк П. К. Моделювання інноваційних процесів для розвитку економіки України. *Прикладні аспекти сучасних міждисциплінарних досліджень*: матеріали I Міжнародної науково-практичної конференції (м. Вінниця, 18 листопада 2022 р.). Вінниця: ДонНУ імені Василя Стуса, 2022. С. 225-228. URL: <https://jpasmd.donnu.edu.ua/article/view/12975>.
49. Корнієнко К. К., Січко Т. В. Сучасні фреймворки та бібліотеки для розробки веб-застосунків. *Прикладні інформаційні технології*: матеріали IV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених (м. Вінниця, 19 травня 2023 р.). Вінниця: ДонНУ імені Василя Стуса, 2023. С. 135-137. URL: <https://jait.donnu.edu.ua/article/view/13988>.

ДОДАТОК А

ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, _____

Повністю вказується ПІБ та статус (посада для працівників, освітня (освітньо-наукова) програма – для здобувачів вищої освіти)

що нижче підписалась/підписався, розуміючи та підтримуючи загальновизнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»;

що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації;

що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволений спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів;

що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

(дата)_____
(підпис)