

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Система керування контентом веб-сайту соціальної мережі

Назва теми

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного  
забезпечення»

Шифр КвРПЗ.190133.19.07.ПЗ

Виконав студент IV курсу, група ПЗ-19-1

  
Підпис

О. О. Зайцев  
Ініціали, прізвище

Керівник канд. техн. наук, доцент  
Науковий ступінь, звання

  
Підпис


І. В. Гурман  
Ініціали, прізвище

Нормоконтролер канд. техн. наук, доцент  
Науковий ступінь, звання

  
Підпис

Ю. В. Форкун  
Ініціали, прізвище

До захисту допускаю:  
Завідувач кафедри інженерії  
програмного забезпечення

  
Підпис

Л. П. Бедратюк  
Ініціали, прізвище

5 06 2023 р.

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри 193

Л. П. Бедратюк

02 01 2023 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЕКТ)

Зайцеву Олегу Олександровичу

Прізвище, ім'я, по батькові студента

1. Тема роботи (проекту) Система керування контентом веб-сайтом соціальної мережі

Керівник роботи (проекту) Іван Васильович Гурман, канд. пед. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Матеріали переддипломної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

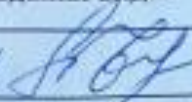
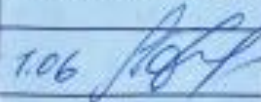


Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація, тестування програмного забезпечення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Презентаційні матеріали (слайди)

Графічна частина (3 діаграми формату А3)

## 6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Форкун Юрій Вікторович, доцент	1.06 	1.06 
Антиплагіат	Гурман Іван Васильович, доцент	 2.06	 2.06

7. Дата видачі завдання « 02 » січня 2023р.

## КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи (проєкту)	Строк виконання етапів проєкту (роботи)	Примітка
1 Ознайомлення з тематикою дипломного проєктування (ДП), визначення та узгодження індивідуальних тем ДП	01.12– 31.12.2022	
2 Збір матеріалу за темою ДП; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2023	
3 Проєктування програмного забезпечення	01.02 – 28.02.2023	
4 Програмна реалізація з використанням відповідних засобів розробки	01.03 – 10.04.2023	
5 Тестування програмного забезпечення	11.04 – 30.04.2023	
6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів	01.05 – 25.05.2023	
7 Попередній захист ДП	Травень 2023 (згідно графіка)	
8 Перевірка ДП на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки	26.05 – 30.05.2023	
9 Здача ДП на кафедру; підготовка ДП для розміщення у репозитарії ХНУ; підготовка до захисту та захист ДП	з 01.06.2023	

Студент



О. О. Зайцев

Підпис

Ініціали, прізвище

Керівник проєкту (роботи)



І. В. Гурман

Підпис

Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система керування контентом веб-сайтом соціальної мережі».

Автор проекту: Зайцев Олег Олександрович.

Керівник проекту: Іван Васильович Гурман.

Пояснювальна записка: 68 с., 11 рис., 1 табл., 7 дод., 41 джерел.

Графічна частина: 12 презентаційних слайдів.

СОЦІАЛЬНА МЕРЕЖА, ВЕБ-ЗАСТОСУНОК, JAVASCRIPT, JSX, MONGODB БАЗА ДАНИХ, CSS

Мета кваліфікаційної роботи: розробка системи керування контентом веб-сайтом соціальної мережі.

У кваліфікаційній роботі проведено аналіз предметної області та її інформаційного забезпечення, визначено вимоги до довідково-інформаційної системи, розроблено її архітектуру та спроектована структура бази даних соціальної мережі.

При розробці програмної системи використано мови програмування JavaScript, JSX, сервер бази даних MongoDB та таблиці стилів CSS.

Практична значимість отриманих результатів полягає у розробці готового до використання програмного продукту у вигляді веб-сайту.

Впровадження системи керування контентом дозволяє оптимізувати роботу соціальної мережі, надає можливість роботи адміністратора з контентом користувачів.

09.06.2023

Дата



Підпис

## ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.190133.19.07.ПЗ	Пояснювальна записка	68		
2	A4		Завдання на кваліфікаційну роботу	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A4		Презентаційні матеріали	12		
5	A3	КвРІПЗ.190133.19.07.E8	UML – діаграма декомпозиції першого рівня	1		
6	A3	КвРІПЗ.190133.19.07.E8	ER – діаграма бази даних	1		
7	A3	КвРІПЗ.190133.19.07.E8	ER – діаграма класів	1		

					<b>КвРІПЗ.190133.19.07.ВД</b>			
Змн.	Арк.	№ докум.	Підпис	Дата	Система керування контентом веб-сайту соціальної мережі.	Літ.	Арк.	Аркуші
Розроб.		Зайцев О.О.		17.06.20			1	1
Перевір.		Гурман І.В.		2.06				
Н. Кантр.		Фаркун Ю.В.		21.06.20				
Затверд.		Бедратюк Л.П.		5.06	Відомість документів	<b>ХНУ, ІПЗ-19-1</b>		

## ЗМІСТ

<b>ВСТУП</b> .....	6
<b>1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	9
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей.....	9
1.2 Аналіз наявного програмно-технічного забезпечення предметної області.....	11
1.3 Визначення функціональних вимог до програмного забезпечення .....	18
<b>2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	20
2.1 Архітектурне проектування .....	20
2.2 Детальне проектування.....	25
2.3 Проектування інтерфейсу користувача .....	36
2.4 Детальне проектування даних.....	38
2.5 Аналіз та вибір технологій розробки клієнтської частини .....	42
2.6 Висновки до другого розділу .....	44
<b>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ</b> .....	45
3.1 Програмна реалізація модулів .....	45
3.2 Керівництво користувача .....	50
3.3 Вимоги до технічних та програмних засобів .....	54
3.4 Вибір та обґрунтування методів тестування веб-застосунку .....	54
3.5 Перевірка на помилки за допомогою unit-тестів .....	58
3.6 Валідація та верифікація веб-застосунку.....	59
3.7 Висновки до другого розділу .....	63
<b>ВИСНОВКИ</b> .....	64
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ</b> .....	65
<b>ДОДАТОК А</b> .....	69

					<b>КвРІПЗ.190133.19.07.ПЗ</b>		
Змін.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Зайцев О.О.		2.06.20	Літ.	Арк.	Акція
Перевір.		Гурман І.В.		2.06		4	68
Н. Контр.		Форкун Ю.В.		2.06.20	<b>ХНУ, ПЗ-19-1</b>		
Затверд.		Бедраток Л.П.		5.06			

Система керування контентом веб-сайту соціальної мережі.

ДОДАТОК Б .....	75
ДОДАТОК В .....	79
ДОДАТОК Г .....	85
ДОДАТОК Д .....	87
ДОДАТОК Е .....	91
ДОДАТОК Є .....	92

## ВСТУП

У сучасному світі соціальні мережі відіграють важливу роль в інтернет-просторі та стали невід'ємною складовою життя суспільства. У зв'язку з цим, все більше компаній та організацій створюють свої веб-сайти на основі соціальних мереж, щоб мати змогу бути на зв'язку зі своїми клієнтами та партнерами, збільшувати продажі та розширювати бізнес.

Соціальні мережі в сучасному світі мають велике значення та широке використання. Ось деякі важливі аспекти та застосування соціальних мереж:

– Спілкування: Соціальні мережі надають можливість спілкуватися з іншими людьми незалежно від їх місця знаходження, що дозволяє зблизити дистанцію між людьми та знайти спільні інтереси.

– Розваги та відпочинок: Соціальні мережі дозволяють відпочити та розважитися, ділившись фотографіями, відео, мемами, музикою та іншими цікавими вмістом.

– Професійна мережа: Багато соціальних мереж також використовуються як інструменти професійної мережі, де можна знайти роботу, встановити професійні контакти, отримати поради та ресурси для кар'єрного зростання.

– Маркетинг та реклама: Багато компаній використовують соціальні мережі як ефективний інструмент маркетингу та реклами, просуваючи свої товари та послуги, взаємодіючи зі своєю аудиторією та підвищуючи свою впізнаваність.

– Засіб новин та інформації: Соціальні мережі стали одним із головних джерел новин та інформації, де люди можуть дізнаватися про події в реальному часі, ділитися новинами та висловлювати свої думки.

– Співпраця та колаборація: Соціальні мережі також дозволяють співпрацювати та колаборувати над проектами з різних куточків світу, спільно вирішувати проблеми та здійснювати творчість.

– Освіта та навчання: Соціальні мережі можуть бути використані в освітніх цілях, де вони можуть допомагати студентам та вчителям спілкуватися,

					КвРІПЗ.190133.19.07.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

співпрацювати, ділитися знаннями та навичками.

– Підтримка та допомога: Соціальні мережі також можуть бути платформами для підтримки та допомоги в різних групах, таких як пацієнти з різними захворюваннями, ветерани, люди з різними викликами та потребами.

– Інновації та розвиток бізнесу: Соціальні мережі можуть сприяти розвитку нових ідей, стартапів та бізнес-проектів. Вони можуть допомагати збирати зворотний зв'язок від аудиторії, проводити дослідження ринку, впроваджувати інноваційні рішення та знаходити нових клієнтів.

Однак, важливо також зазначити, що соціальні мережі також можуть мати свої виклики, такі як проблеми з приватністю, негативним впливом на здоров'я та добробут людей, поширенням фейкової інформації, кібербулінгом та іншими негативними явищами. Тому важливо використовувати соціальні мережі відповідально, зберігаючи свою конфіденційність та здоровий глуздом.

Тому важливо розуміти, що використання соціальних мереж вимагає відповідального підходу. Для того, щоб максимально використовувати їх позитивний потенціал, варто бути обізнаним інформаційно, встановлювати налаштування приватності, бути критичним до отриманої інформації та поводитися в мережі з повагою до інших користувачів. Взаємодія в соціальних мережах може бути корисною, але також важливо бути свідомим та критичним споживачем інформації, а також забезпечувати свою власну кібербезпеку та ментальне здоров'я.

Для того, щоб забезпечити ефективну роботу веб-сайту соціальної мережі, необхідна система керування контентом (CMS), яка дозволяє зберігати, оновлювати та керувати вмістом сайту. Використання CMS є ключовим фактором у розвитку веб-сайту соціальної мережі та впливає на його функціональність та безпеку.

У ході роботи будуть розглянуті такі питання, як функції CMS соціальної мережі, інструменти та технології, що використовуються для її розробки, аналіз існуючих CMS, розробка власної CMS, а також тестування та оцінка її ефективності.

Актуальність роботи полягає у необхідності розробки високоефективних та

					<i>КеРІПЗ.190133.19.07.ПЗ</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

безпечних CMS для соціальних мереж, що забезпечує стабільну та ефективну роботу веб-сайтів соціальних мереж та покращує їх функціональність. Результати дослідження та розробки нової CMS для соціальної мережі можуть стати корисними для компаній та організацій, які планують розробити свій веб-сайт на основі соціальної мережі. Нова CMS може забезпечити їм ефективну та безпечну роботу сайту, що позитивно позначиться на розвитку їх бізнесу.

Крім того, розробка нової CMS для соціальної мережі може внести свій вклад у загальний розвиток інтернет-технологій та поліпшення інтернет-екосистеми. Результати дослідження та розробки нової CMS можуть стати цінним внеском в академічну спільноту та відкриті для усіх користувачів.

Отже, тема "Система керування контентом веб-сайту соціальної мережі" є актуальною та має значний практичний та теоретичний потенціал. Дослідження та розробка нової CMS можуть позитивно вплинути на розвиток інтернет-простору та веб-сайтів соціальних мереж, що забезпечить покращення функціональності та безпеки їх роботи.

Метою проекту є дослідження та аналіз існуючих соціальних мереж та CMS для них, визначення їх переваг та недоліків, розробка власної CMS для соціальної мережі, а також тестування та оцінка її ефективності.

Об'єкт дослідження – процес організації роботи та взаємодії адміністратора та користувачів за допомогою систем керування контентом соціальної мережі.

Предмет – створення системи керування контентом для соціальної мережі.

Виходячи із мети, об'єкту та предмету дослідження можна окреслити завдання на кваліфікаційну роботу:

- дослідити предметну область;
- здійснити аналіз існуючих програмних рішень даної галузі;
- розробити технічне завдання;
- спроектувати розробку програмного забезпечення;
- розробити інструкцію користувача.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						9
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

У сучасному світі соціальні мережі стали невід'ємною частиною нашого життя. Вони є популярними майданчиками для спілкування, обміну інформацією та взаємодії з людьми з усього світу. Соціальні мережі є важливими інструментами для бізнесу, маркетингу, політики, науки та інших галузей. У зв'язку з цим, важливо розуміти структурні та функціональні особливості соціальних мереж для того, щоб ефективно використовувати їх потенціал та забезпечувати безпеку користувачів.

Щоб соціальна мережа набула популярності потрібно, щоб люди зацікавились нею потрібно розробити унікальну та привабливу концепцію: Перед запуском соціальної мережі потрібно визначити, яку проблему вона вирішує або яку потребу задовольняє. Також важливо зрозуміти, хто є цільовою аудиторією та які її потреби.

Також важливо забезпечити зручний та легкий інтерфейс: Люди люблять користуватися соціальними мережами, які мають зручний та легкий інтерфейс. Тому важливо забезпечити зручну навігацію, прості форми реєстрації та входу, зручне розміщення контенту та інші важливі елементи інтерфейсу.

Потрібно забезпечити безпеку та конфіденційність: користувачі соціальних мереж цінують конфіденційність та безпеку своїх даних. Тому важливо забезпечити надійність та безпеку веб-сайту, захист від хакерів та зловмисників, а також забезпечити конфіденційність персональних даних.

Для успішності соціальної мережі потрібно забезпечувати функціональність та інновації: користувачі люблять соціальні мережі, які постійно розвиваються та вдосконалюються. Тому важливо забезпечувати функціональність та інновації, які зроблять соціальну мережу більш цікавою та корисною для користувачів. Наприклад, додаткові функції, зручні інструменти, інтерактивність та інші.

Контент є ключовим елементом популярності соціальної мережі. Важливо надавати користувачам цікавий та корисний контент, який спонукає їх до

					<b>КвРІПЗ.190133.19.07.ПЗ</b>	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

повернення на сайт. Це може бути різноманітний контент: відео, фото, текстові повідомлення, аудіозаписи, статті, новини та інше. Важливо, щоб контент був актуальним, оригінальним та привабливим для цільової аудиторії.

Цільова аудиторія соціальної мережі - це зазвичай молоді люди віком від 18 до 34 років, а також молодші підлітки віком від 13 до 17 років. Ця аудиторія переважно складається з користувачів зі середнім та високим рівнем доходів, які мають доступ до мобільних пристроїв та інтернету.

Спілкування в соціальних мережах стало важливою складовою нашого сучасного життя. Соціальні мережі дають можливість людям з усього світу зв'язуватися між собою, обмінюватися інформацією, думками, ідеями та досвідом.

Однією з важливих переваг спілкування в соціальних мережах є зручність і доступність. Люди можуть спілкуватися між собою в будь-який час та в будь-якому місці з допомогою смартфонів, планшетів або комп'ютерів.

Крім того, соціальні мережі дають можливість швидко та ефективно обмінюватися інформацією. Наприклад, важливі новини можуть поширюватися по мережі зі швидкістю світла, дозволяючи людям вчасно отримувати необхідну інформацію.

Соціальні мережі також дозволяють знаходити та підтримувати зв'язки зі старими та новими друзями, колегами, рідними. Завдяки соціальним мережам люди можуть залишатися на зв'язку з тими, хто знаходиться далеко від них, що допомагає зменшувати відстань та зберігати дружбу чи знайомство на тривалий час.

Соціальна мережа складається зі спільноти людей, які взаємодіють один з одним, обмінюються інформацією та спільно діють у певній тематичній області.

Профілі користувачів є основною складовою соціальної мережі. Вони містять особисту інформацію про користувача, таку як ім'я, вік, місцезнаходження, професійні дані та інтереси. Користувачі можуть додавати друзів та підписуватися на профілі інших користувачів, щоб бути в курсі їхньої активності в соціальній мережі.

Групи є ще однією важливою складовою соціальної мережі. Вони можуть

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						11
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

бути створені користувачами з різних причин, наприклад, для обговорення спільних інтересів, подій або для співпраці в професійній сфері. Групи можуть бути відкритими або закритими, іноді потрібно одержати дозвіл на вступ до них. Користувачі можуть додавати пости, коментарі та ділитися інформацією в групах.

Сторінки можуть бути створені компаніями, брендами або публічними особистостями для спілкування зі своїми фанами та клієнтами. Сторінки можуть містити інформацію про компанію, її продукти та послуги, новини та інші матеріали.

Для підтримки взаємодії між користувачами соціальна мережа може надавати додаткові функції, такі як повідомлення, коментарі, лайки та репости. Користувачі можуть обговорювати публікації, додавати коментарі та лайки, а також поширювати контент, який їм сподобався.

Крім цього, соціальна мережа може мати різні функції, які допомагають користувачам знаходити нових друзів та підписки, такі як пошук за інтересами, рекомендації та знайомства через спільних друзів.

## 1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Соціальні мережі є невід'ємною частиною нашого життя, і має досить вагомий вплив на наше суспільство. Такі соціальні мережі як LinkedIn, Twitter, куточків планети. Використовуючи ці соціальні мережі можна ділитись фото, відео ресурсами а також різноманітною інформацією.

- це найбільша соціальна мережа, яка налічує понад 2,8 мільярдів активних користувачів у всьому світі. Функціонал цієї мережі дозволяє створювати профілі, додавати друзів, ділитися контентом, приєднуватися до груп і спілкуватися в чаті. Крім цього, Facebook надає можливість рекламувати свої товари та послуги, а також продавати їх на цій платформі.

Переваги соціальної мережі Facebook описані далі.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Facebook, зі своїми понад 2 мільярдами активних користувачів по всьому світу, є однією з найбільших соціальних мереж, що дозволяє компаніям та брендам залучати більше клієнтів та збільшувати свою аудиторію.

Крім того, Facebook надає можливість створювати таргетовану рекламу, яка дозволяє рекламодавцям детально налаштовувати свої рекламні кампанії та дотримуватися бюджету.

Facebook має велику кількість функцій та інструментів, що дозволяють користувачам спілкуватися, ділитися контентом, групуватися за інтересами, відстежувати новини та багато іншого.

Крім того, Facebook надає користувачам можливість відстежувати показники взаємодії з аудиторією, такі як кількість лайків, коментарів, репостів та інших взаємодій з контентом.

Нарешті, Facebook має додатки для мобільних пристроїв, що дозволяє користувачам залишатися на зв'язку та використовувати платформу з будь-якого місця та в будь-який час.

Недоліки соціальної мережі Facebook:

– Питання конфіденційності. Facebook був критикований за порушення приватності користувачів та збір їхніх даних без їхньої згоди.

– Залежність від алгоритмів. Використання алгоритмів Facebook для відбору та показу контенту може привести до створення "еко-камер" та спотворення поглядів та думок користувачів, що може вплинути на вибори та громадську думку.

– Реклама та спам. Перенасиченість реклами та спаму на Facebook може стати нав'язливим для користувачів та негативно вплинути на їхній досвід використання платформи.

– Небезпека для дітей та підлітків. Facebook може бути небезпечним для дітей та підлітків, які можуть зіткнутися з різними загрозами в Інтернеті, такими як кібербулінг та експлуатація.

Twitter - це соціальна мережа та мікроблогінгова платформа, на якій

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

користувачі можуть публікувати короткі повідомлення (твіти) з обмеженням у 280 символів. Такі повідомлення можуть містити тексти, фотографії, відео та інші медіафайли. Користувачі можуть підписуватися на інших користувачів та отримувати їхні оновлення у своїх новинах. Twitter використовується для спілкування, отримання новин та інформації, реклами, маркетингу та багато іншого. Він є однією з найпопулярніших соціальних мереж у світі з більш ніж 330 мільйонами активних користувачів.

#### Переваги соціальної мережі Twitter:

- Швидке та просте спілкування: Twitter надає можливість легко та швидко спілкуватися зі своїми друзями, колегами та іншими користувачами з усього світу.
- Велика аудиторія: Twitter має понад 330 мільйонів активних користувачів з усього світу, що дозволяє користувачам ділитися своїми думками та ідеями з широкою аудиторією.
- Інформаційний потік: Twitter є важливим джерелом новин та інформації з усього світу. Користувачі можуть отримувати актуальну інформацію про події, теми та новини, які їх цікавлять.
- Взаємодія з публічними особами: Twitter є майданчиком для взаємодії з публічними особами, які можуть використовувати платформу для спілкування зі своїми прихильниками та критиками.
- Використання хештегів: Twitter використовує хештеги, які допомагають користувачам швидко знайти теми, які їх цікавлять, а також ділитися своїми думками з великою аудиторією.

#### Недоліки соціальної мережі Twitter:

- Обмеження довжини повідомлень, яке може ускладнити детальне ділення інформації;
- Ризик стати ехо-камерою та зміщенням думок, що є загальною проблемою соціальних мереж;
- Велика кількість інформації, яка може призводити до проблеми з вибором цікавої та корисної інформації;

					КвРІПЗ.190133.19.07.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

- Ризик від негативних та токсичних коментарів, які можуть відлякувати користувачів від публічного обговорення важливих тем;
- Можливість поширення неперевіреної або фейкової інформації через віртуальний та швидкий характер соціальної мережі.

Інстаграм (Instagram) - це популярна мобільна соціальна мережа, яка дозволяє користувачам ділитися фотографіями та відео, а також обмінюватися повідомленнями. Платформа була запущена в 2010 році і придбана Facebook у 2012 році. Зараз Інстаграм має понад 1 мільярд активних користувачів, що робить його однією з найбільших соціальних мереж у світі. Крім публікації зображень та відео, на Інстаграмі можна створювати інтерактивні сторіс, додавати фільтри та ефекти, а також слідкувати за іншими користувачами та брендами.

#### Переваги соціальної мережі Instagram:

- Візуальний контент: Instagram - це переважно візуальна соціальна мережа, яка дозволяє користувачам ділитися фотографіями та відео. Це робить її ідеальним майданчиком для візуального маркетингу, де компанії можуть відобразити свої продукти та послуги у привабливому візуальному вигляді.
- Активність аудиторії: Instagram є однією з найбільш активних соціальних мереж, що дозволяє користувачам взаємодіяти один з одним та спілкуватися на різних рівнях, включаючи коментарі, приватні повідомлення, а також поділитися відгуками.
- Можливості реклами: Instagram надає можливості для маркетингу продуктів та послуг через різні види реклами, такі як рекламні пости, історії, реклама в IGTV та більше.
- Глобальна аудиторія: Загальна аудиторія Instagram налічує більше мільярда користувачів у всьому світі, що дає можливість компаніям та брендам привернути увагу аудиторії з усього світу.
- Аналітика: Instagram надає користувачам та бізнес-акаунтам доступ до детальної аналітики, що дозволяє відстежувати ефективність реклами та взаємодії з аудиторією.

					<b>КвРІПЗ.190133.19.07.ПЗ</b>	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

– Інфлюенсер-маркетинг: Instagram є ключовим майданчиком для інфлюенсер-маркетингу, де користувачі можуть співпрацювати зі знаменитостями та впливовими особистостями, щоб привернути увагу до своїх продуктів або послуг.

– Легкість використання: Instagram має досить простий та легкий в інтерфейс.

Недоліки соціальної мережі Instagram:

– Алгоритм обмеженого охоплення: Instagram використовує складні алгоритми, які визначають, кому будуть показуватися ваші дописи. Це може призвести до того, що ваш контент не досягне вашої аудиторії.

– Залежність від візуального змісту: Instagram - це переважно візуальна соціальна мережа, тому користувачам, які прагнуть ділитися текстовим контентом або не володіють фотографічним мистецтвом, може бути складно побудувати свої акаунти.

– Негативність та токсичність: Як і більшість соціальних мереж, Instagram може стати місцем для поширення негативних і токсичних коментарів. Крім того, негативна репутація або коментар може вплинути на психологічний стан користувача.

– Залежність від вподобань та коментарів: Instagram може стати місцем залежності від вподобань та коментарів. Користувачі можуть відчувати незадоволення, якщо їхні дописи не отримують достатньо вподобань або коментарів, що може вплинути на їхню самооцінку.

– Маніпуляція фотографіями: Завдяки різноманітним фільтрам та редагувальним інструментам користувачі можуть легко змінювати свої фотографії. Це може призвести до створення фотографічного ідеалу, який не відображає реальності.

LinkedIn - це соціальна мережа для професійного спілкування, що заснована в 2002 році. Ця платформа дозволяє користувачам знайомитися з професійними контактами, шукати роботу або працівників, публікувати вакансії, статті та інші

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

матеріали, пов'язані з робочими справами. LinkedIn також дозволяє користувачам приєднуватися до груп за інтересами та обговорювати різноманітні теми зі своїми колегами. Сьогодні LinkedIn є однією з найбільших та найпопулярніших соціальних мереж для професійних мережування.

#### Переваги соціальної мережі LinkedIn:

– Професійна мережа: LinkedIn є основним соціальним медіа для професіоналів і бізнесу. Вона дозволяє користувачам знайомитися з людьми зі свого галузі та встановлювати бізнес-контакти.

– Розширення можливостей кар'єри: LinkedIn є важливим інструментом для пошуку роботи та збільшення своїх можливостей кар'єри. Користувачі можуть створювати свій профіль, додавати до нього свої досягнення та навички, отримувати рекомендації від колег та співробітників, шукати вакансії та знаходити підходящих роботодавців.

– Можливості маркетингу та реклами: LinkedIn надає можливості для маркетингу та реклами, особливо для бізнес-користувачів. Вони можуть створювати компанійські сторінки, публікувати статті та контент, розміщувати оголошення про роботу, рекламувати свої послуги та товари та залучати нових клієнтів.

– Розвиток освіти та знань: LinkedIn має багато корисних матеріалів для розвитку навичок та знань у різних галузях. Користувачі можуть брати участь у вебінарах, онлайн-курсах, отримувати новини та іншу корисну інформацію, що допомагає розвивати свою кар'єру та компетенції.

– Безпека та конфіденційність: LinkedIn надає високий рівень захисту користувачів та їхньої інформації. Вона має політику конфіденційності та безпеки, яка забезпечує захист від шахрайства, спаму та інших небезпек.

#### Недоліки соціальної мережі LinkedIn:

– Надмірна реклама: LinkedIn може бути досить рекламним, з багатьма спонсорованими оголошеннями і рекламою на сторінках користувачів. Це може бути дратівливим для користувачів, які хочуть просто спілкуватися з колегами і

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

знайомитися з новими людьми.

– Обмежені функції: LinkedIn не має такої широкої функціональності, як Facebook або Twitter. Наприклад, на LinkedIn немає можливості створення подій або створення власного блогу. Крім того, деякі функції, такі як повідомлення, можуть бути обмежені в безкоштовній версії.

– Сильна залежність від приватності: LinkedIn має більш сувору політику щодо захисту приватності, ніж деякі інші соціальні мережі. Наприклад, LinkedIn забороняє стеження за профілями інших користувачів, якщо ви не знаєте їх особисто. Це може бути обмеженням для тих, хто хоче збирати інформацію про своїх конкурентів або потенційних роботодавців.

– Бізнес-фокус: LinkedIn зазвичай більше спрямований на професійне спілкування і мережевий маркетинг, ніж на особисте спілкування та розваги. Це може бути обмеженням для тих, хто хоче використовувати соціальну мережу для розваг або для особистого спілкування з друзями та родичами.

– Ризик для кар'єри: LinkedIn може бути корисним для знаходження нових робочих місць і контактів, але в той же час, якщо ви необачні з тим, що публікуєте на своєму профілі, це може пошкодити вашу репутацію. Наприклад, якщо ви ділитесь непристойними або неадекватними думками, це може вплинути на вашу можливість отримати роботу або зберегти її, особливо якщо ваш роботодавець вирішить перевірити ваш профіль в LinkedIn.

– Ризик кібербезпеки: LinkedIn може бути потенційною мішенню для кіберзлочинців, які шукають доступ до конфіденційної інформації або особистих даних користувачів. Хакери можуть намагатися зламати профілі LinkedIn або використовувати соціальну мережу для фішингу або інших видів кібератак.

Загальний вигляд проаналізованих соціальних мереж можна побачити у додатку Б.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						18
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

### 1.3 Визначення функціональних вимог до програмного забезпечення

Що має бути в нашій соціальній мережі?

Перше, що має побачити користувач, який зайшов у соціальну мережу це вікно реєстрації. Під час реєстрації новий користувач повинен ввести свою унікальну електронну пошту та придумати пароль. Після реєстрації користувач буде мати доступ до повного функціоналу сайту, якщо авторизується. Після авторизації користувач матиме змогу переглядати сторінки інших людей, поширювати дописи, коментувати їх а також оцінювати. Також передбачене листування, перегляд новин.

Основними вимогами до ПЗ є:

– Масштабованість: програма повинна мати можливість розширюватися та масштабуватися, щоб задовольняти потреби зростаючої кількості користувачів та забезпечувати необхідну функціональність.

– Безпека: програма має забезпечувати безпеку даних користувачів за допомогою різних методів захисту, таких як шифрування та двофакторна автентифікація.

– Функціональність: програма повинна мати набір функцій, які дозволяють користувачам розмішувати, редагувати та взаємодіяти з контентом, а також додавати друзів, створювати та приєднуватися до груп, мереж підприємств та інших.

– Мобільність: програма повинна бути оптимізована для мобільних пристроїв, щоб забезпечити доступ до соціальної мережі з будь-якого місця та в будь-який час.

– Аналітика та статистика: програмне забезпечення повинно давати можливість відслідковувати та аналізувати показники взаємодії користувачів з контентом з метою поліпшення маркетингових кампаній та забезпечення кращої комунікації з аудиторією.

– Інтеграція з іншими сервісами: програмне забезпечення має підтримувати

					КвРІПЗ.190133.19.07.ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

можливість інтеграції з іншими соціальними мережами та сервісами, щоб забезпечити більшу функціональність та зручність для користувачів.

– Підтримка користувачів: програмне забезпечення повинно забезпечувати належну підтримку користувачів, включаючи можливість звертатися до технічної підтримки та вирішувати проблеми оперативно та ефективно. Крім того, програмне забезпечення повинно допомагати користувачам налаштовувати облікові записи, приватність та інші важливі параметри.

– Швидкість та продуктивність: щоб забезпечити користувачам зручну та ефективну роботу з соціальною мережею, програмне забезпечення повинно бути швидким та продуктивним.

– Надійність та стабільність: для запобігання втрати даних та забезпечення постійного доступу до соціальної мережі, програмне забезпечення повинно бути надійним та стабільним.

– Автоматичне оновлення: щоб запобігти появі проблем, пов'язаних з застарілими версіями програмного забезпечення та забезпечити користувачам доступ до останніх версій соціальної мережі, програмне забезпечення повинно мати можливість автоматичного оновлення.

– Сумісність з різними платформами: щоб забезпечити максимальну доступність та зручність взаємодії користувачів з соціальною мережею, програмне забезпечення повинно бути сумісним з різними платформами, такими як веб-браузери, мобільні пристрої та інші.

У короткому висновку можна сказати, що соціальна мережа повинна мати функціонал, який дозволяє користувачам зареєструватися, авторизуватися та взаємодіяти з іншими користувачами шляхом розміщення дописів, коментування, оцінювання та листування. Вона повинна бути масштабованою, безпечною, функціональною, мобільною, з аналітикою та статистикою, інтеграцією з іншими сервісами, підтримкою користувачів, швидкістю та продуктивністю, надійністю та стабільністю, підтримкою різних мов, автоматичним оновленням та сумісністю з різними платформами.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Архітектурне проектування

Клієнт-серверна архітектура - це модель взаємодії між комп'ютерними програмами або системами, в якій клієнтські програми взаємодіють з сервером для отримання ресурсів та послуг. У випадку соціальних мереж, клієнтська частина - це інтерфейс користувача, який працює на стороні користувача, тоді як серверна частина забезпечує зберігання та обробку даних, відправку та отримання повідомлень, керування безпекою та багато іншого.

Існує кілька видів архітектур взаємодії між клієнтом та сервером. Основні типи клієнт-серверних архітектур включають:

– Клієнт-сервер з централізованою обробкою: у цій архітектурі клієнти взаємодіють з одним центральним сервером, який відповідає за обробку всіх запитів та відповідей. Цей підхід дозволяє зменшити складність мережі, але може створювати єдину точку відмови.

– Клієнт-сервер з розподіленою обробкою: у цій архітектурі клієнти взаємодіють з багатьма серверами, які розподілені по всій мережі. Кожен сервер відповідає за обробку певної категорії запитів та відповідей. Цей підхід дозволяє зменшити ризик виникнення єдиної точки відмови та підвищити масштабованість системи.

– Peer-to-peer архітектура: у цій архітектурі всі клієнти є рівноправними вузлами мережі, які можуть взаємодіяти між собою без посередництва сервера. Цей підхід дозволяє зменшити навантаження на сервер та підвищити стійкість системи до відмов.

– Клієнт-сервер з кешуванням: у цій архітектурі сервер зберігає кешовану копію даних, що найчастіше запитуються клієнтами. Клієнти спочатку запитують дані у кеші сервера, що дозволяє зменшити навантаження на сервер та знизити час відповіді на запит.

Вибір архітектури залежить від конкретних потреб системи, а також від

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						21
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

характеристик мережі та застосунків, які будуть взаємодіяти між собою. Наприклад, якщо система має бути високонавантаженою, то краще використовувати клієнт-сервер з розподіленою обробкою, щоб рівномірно розподіляти навантаження між серверами. Якщо система має працювати в умовах обмежених ресурсів, то краще використовувати клієнт-сервер з кешуванням, щоб зменшити навантаження на сервер та знизити час відповіді на запит. Вибір архітектури також може залежати від функціональних вимог системи та рівня безпеки, що потрібен для системи.

У клієнт-серверній архітектурі з централізованою обробкою, клієнти взаємодіють з одним центральним сервером, який відповідає за обробку всіх запитів та відповідей. Основна ідея полягає в тому, що сервер зберігає та керує всіма даними та ресурсами, які потрібні клієнтам, тоді як клієнти забезпечують лише інтерфейс для взаємодії з цими ресурсами.

Клієнти можуть відправляти запити до сервера, які містять інформацію про те, які дані потрібні, а сервер обробляє ці запити та повертає відповіді, що містять запитані дані. Сервер може також обробляти та зберігати дані, які надходять від клієнтів.

Одним з головних переваг цієї архітектури є те, що вона забезпечує централізоване керування та контроль над даними та ресурсами, що полегшує адміністрування системи та зменшує складність мережі. Крім того, цей підхід може бути досить ефективним для менших мереж з невеликою кількістю клієнтів, коли обсяг даних та навантаження на сервер не є дуже великим.

Однак, цей підхід може мати також деякі недоліки. Наприклад, при використанні цієї архітектури може створюватися єдина точка відмови, тобто якщо сервер виходить з ладу, то всі клієнти втрачають доступ до даних. Також, збільшення кількості клієнтів та даних може призвести до збільшення навантаження на сервер та зниження продуктивності.

Найефективнішою є клієнт-сервер з централізованою обробкою, де всі запити від користувачів обробляються центральними серверами. Така архітектура

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						22
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

дозволяє контролювати весь обмін даними між клієнтами та серверами, що забезпечує підвищену безпеку та конфіденційність даних користувачів. Крім того, централізований підхід дозволяє забезпечити швидку та стабільну роботу своєї соціальної мережі з мільярдами користувачів.

Клієнтська частина соціальної мережі складається з веб-сторінок та мобільних застосунків, які дозволяють користувачам взаємодіяти з сервісом. Інтерфейс користувача забезпечує зручний та ефективний спосіб взаємодії з соціальною мережею, що дозволяє користувачам швидко знаходити потрібну інформацію, спілкуватися з іншими користувачами та виконувати інші завдання.

Інтерфейс користувача є дуже важливою частиною клієнтської частини соціальної мережі. Його метою є забезпечення зручного та простого використання сервісу користувачами. Інтерфейс повинен бути досить простим та інтуїтивно зрозумілим, щоб користувачі могли швидко знайти необхідну інформацію, створити або переглянути записи, додавати коментарі та відгуки, спілкуватися з іншими користувачами та виконувати інші дії.

Для досягнення цих цілей інтерфейс користувача повинен мати чітко визначену структуру та логіку взаємодії з користувачем. Зазвичай це включає в себе розташування елементів керування та навігації, відображення інформації та контенту, роботу зі списками та фільтрами, взаємодію зі сторонніми сервісами та багато іншого.

Окрім того, інтерфейс користувача повинен бути адаптивним до різних пристроїв та розмірів екранів, оскільки користувачі можуть використовувати соціальну мережу на різних пристроях, включаючи комп'ютери, планшети та смартфони. Для забезпечення зручності та ефективності взаємодії з користувачами, інтерфейс також повинен бути естетичним та привабливим з точки зору дизайну та візуальної привабливості.

Серверна частина соціальної мережі складається з компонентів, які забезпечують обробку запитів користувачів та зберігання даних. Серверна частина може бути розподілена на кілька фізичних серверів, які працюють у режимі

					<i>KePIПЗ.190133.19.07.ПЗ</i>	Арк.
						23
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

кластера. Для забезпечення швидкодії та високої доступності, серверна частина може використовувати різні технології та протоколи, такі як TCP/IP, HTTP, REST, JSON та інші.

Серверна частина соціальної мережі може складатися з декількох компонентів, таких як веб-сервери, бази даних, кеш-сервери та інші. Кожен з цих компонентів виконує свої функції і взаємодіє з іншими компонентами, щоб забезпечити роботу соціальної мережі.

Веб-сервери забезпечують обробку запитів користувачів, відправку веб-сторінок та іншого контенту. Бази даних забезпечують зберігання даних користувачів, таких як профілі, повідомлення, фотографії та інші. Кеш-сервери забезпечують швидкий доступ до даних, що часто використовуються, зменшуючи навантаження на базу даних.

TCP/IP є основним протоколом, який використовується для забезпечення комунікації між серверами. HTTP є протоколом, який використовується для передачі веб-сторінок та іншого контенту між сервером та клієнтом. REST та JSON є технологіями, які дозволяють розробникам створювати API, що можуть використовуватися для обміну даними між серверною та клієнтською частинами. Ці технології дозволяють забезпечувати ефективну та швидку роботу соціальної мережі, а також забезпечити її доступність для користувачів у будь-який час та з будь-якого місця.

Взаємодія між клієнтською та серверною частинами забезпечується за допомогою мережі. Клієнти можуть відправляти запити до сервера та отримувати відповіді, а сервер може відправляти повідомлення клієнтам, які підписалися на отримання сповіщень. Для забезпечення безпеки передачі даних між клієнтською та серверною частинами можуть використовуватися різні протоколи шифрування та аутентифікації, такі як SSL, TLS, OAuth та інші.

Безпека передачі даних є дуже важливим аспектом взаємодії між клієнтською та серверною частинами. Протоколи шифрування, такі як SSL (Secure Sockets Layer) та його наступник TLS (Transport Layer Security), дозволяють захистити

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						24
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

передачу даних від несанкціонованого доступу та перехоплення.

OAuth (Open Authorization) - протокол авторизації, який дозволяє користувачам використовувати свої облікові записи на одному веб-сайті для входу на інший веб-сайт без необхідності вводити свої дані для авторизації. Це дозволяє зменшити ризик злому облікового запису користувача на іншому веб-сайті, оскільки не потрібно вводити свій пароль на цьому сайті.

Інші протоколи безпеки можуть включати в себе механізми аутентифікації, авторизації, шифрування, контролю доступу та перевірки цілісності даних. Використання таких протоколів дозволяє забезпечити безпеку передачі даних та захист від різних видів атак.

Клієнт-серверна архітектура дозволяє соціальним мережам забезпечити швидку та ефективну взаємодію з користувачами та забезпечити безпеку та надійність системи. Додатково, така архітектура дозволяє легко масштабувати систему при збільшенні обсягу даних та навантаження на сервіс.

Серверна архітектура є дуже складною та розгалуженою, і складається з багатьох компонентів, які працюють разом для забезпечення функціональності соціальної мережі. Основні компоненти серверної архітектури включають:

– Frontend-сервери: ці сервери відповідають за обробку запитів користувачів та надсилання відповідей назад до клієнта. Зазвичай використовують технологію React для побудови свого інтерфейсу користувача.

– Back-end сервіси: ці сервіси відповідають за обробку бізнес-логіки та забезпечення функціональності соціальної мережі, такої як створення повідомлень, додавання друзів, завантаження фотографій та відео і т. д. Зазвичай використовують мікросервісну архітектуру для своїх back-end сервісів.

– Бази даних: зазвичай використовують розподілену базу даних, яка дозволяє розподіляти навантаження між різними серверами та забезпечувати швидкий доступ до даних. База даних соціальної мережі побудована на основі технології MongoDB.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2.2 Детальне проектування

Node.js є платформою, що дозволяє виконувати JavaScript-код на стороні сервера, а не лише в браузері. Завдяки цьому, Node.js може бути використаний як серверний фреймворк для побудови веб-застосунків.

Node.js має декілька переваг та особливостей використання, включаючи:

- Швидкість та ефективність: Node.js працює на основі неблокуючого, асинхронного вводу/виводу, що дозволяє обробляти більше запитів одночасно та зменшує час відповіді сервера.

- Спільне використання мови програмування: Node.js використовує JavaScript, що дозволяє розробникам використовувати одну мову як на стороні сервера, так і на стороні клієнта. Це зменшує необхідність вивчати багато мов програмування та сприяє більш простому розробленню та підтримці коду.

- Наявність величезної кількості пакетів: Node.js має широкий спектр пакетів та модулів, які можуть значно полегшити розробку та забезпечити додаток більшою функціональністю.

- Велика спільнота розробників: Node.js має велику та активну спільноту розробників, яка забезпечує підтримку, вирішення проблем та розширення можливостей платформи.

- Сумісність зі стандартами веб-розробки: Node.js забезпечує зручну роботу з HTTP-протоколом та дозволяє розробникам легко створювати веб-застосунки з використанням стандартних інструментів та технологій.

- Підтримка складних застосунків: Node.js дозволяє розробникам створювати складні веб-застосунки, що містять багато компонентів та можуть масштабуватися на багато серверів.

Однак, Node.js має деякі недоліки, такі як:

- Потребує високої кваліфікації: Node.js може бути складним для вивчення та використання для розробників з малою кількістю досвіду в програмуванні та веб-розробці.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

– Не підходить для важких обчислень: Node.js не є оптимальним вибором для важких обчислень, так як воно не підтримує багатопоточність.

– Потребує додаткових інструментів для роботи з базами даних: Node.js не має вбудованої підтримки для багатьох типів баз даних, тому для роботи з ними необхідно використовувати додаткові інструменти та бібліотеки.

– Хоча Node.js не є універсальним інструментом, воно є популярним вибором для створення веб- застосунків, особливо з невеликою кількістю складних обчислень та інтенсивним вводом/виводом. Він може бути корисним для команди розробників, що володіють досвідом в JavaScript та веб-розробці, та шукають швидкий та ефективний спосіб розробки веб- застосунків.

React та Node.js можуть використовуватися разом для створення повноцінного веб- застосунку. React може використовуватися для створення клієнтської частини застосунку, тобто користувацького інтерфейсу. Node.js може використовуватися для створення серверної частини застосунку, тобто логіки, яка забезпечує зв'язок з базою даних та обробку запитів користувачів.

React та Node.js можуть взаємодіяти між собою за допомогою API, яке забезпечує зв'язок між клієнтською та серверною частинами. API дозволяє передавати дані між React та Node.js та взаємодіяти з базою даних, щоб забезпечити потрібну функціональність застосунку.

API (інтерфейс програмування додатків) - це спосіб, за допомогою якого різні програми, сервіси та системи можуть обмінюватись даними між собою. API забезпечує можливість запитувати та отримувати дані з інших додатків чи сервісів, використовуючи визначений набір правил та протоколів.

API можуть мати різні формати взаємодії, наприклад REST (Representational State Transfer), SOAP (Simple Object Access Protocol), GraphQL та інші. Кожен формат має свої переваги та обмеження, та вибір залежить від конкретних потреб проекту.

Розробники можуть використовувати готові API, щоб отримувати дані з різних джерел, або створювати власні API, щоб надавати доступ до своїх даних та

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

функціональності іншим додаткам. Це дозволяє розширювати можливості програм, підвищувати їхню ефективність та забезпечувати взаємодію між різними сервісами та системами.

Існує багато видів баз даних, і кожен з них має свої особливості та використання, в залежності від потреб застосунку. Основні види баз даних включають:

– Реляційні бази даних: це найбільш поширений тип баз даних. Дані зберігаються у вигляді таблиць з рядками та стовпцями, які можуть бути пов'язані між собою за допомогою ключів. Прикладами реляційних баз даних є MySQL, PostgreSQL та Oracle.

– Нереляційні бази даних: також відомі як NoSQL бази даних, ці бази даних зберігають дані у форматі, що не відповідає традиційній таблиці. Вони можуть використовуватись для зберігання документів, графів, ключ-значення та інших форматів даних. Приклади NoSQL баз даних включають MongoDB, Cassandra та Redis.

Нереляційні бази даних: це бази даних, де дані зберігаються в іншому форматі, ніж таблиці з реляційними базами даних. Нереляційні бази даних зазвичай використовуються для зберігання великих обсягів даних, таких як графи, документи та масиви. Прикладами нереляційних баз даних є MongoDB, Cassandra та Couchbase.

MongoDB: це документ-орієнтована база даних, яка зберігає дані у вигляді документів, що містять пари ключ-значення. MongoDB може бути використана як для масштабних веб-застосунків, так і для простих проектів.

Основні переваги нереляційних баз даних (NoSQL) над реляційними базами даних (RDBMS) полягають в їх гнучкості, масштабованості та високій продуктивності. Деякі з основних переваг NoSQL баз даних в порівнянні з RDBMS:

– Гнучкість: NoSQL бази даних дозволяють зберігати дані будь-якого типу та розміру. Вони не мають фіксованих схем даних, що дає можливість зберігати дані в форматі, що відповідає вашим потребам. Це дозволяє додавати нові поля до

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						28
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

документів без перекомпіляції бази даних.

– Масштабованість: NoSQL бази даних можуть бути легко масштабовані, дозволяючи зберігати та обробляти великі обсяги даних. Ви можете розширити кластер NoSQL бази даних, додавши нові вузли, щоб підвищити продуктивність та надійність.

– Продуктивність: NoSQL бази даних можуть забезпечити високу продуктивність завдяки своїй архітектурі, яка дозволяє швидко зберігати та виконувати запити до даних. Вони можуть бути швидшими за RDBMS, коли потрібна обробка великої кількості даних.

– Швидкість розробки: NoSQL бази даних можуть бути легко розгорнуті та використані для розробки застосунку. Вони не вимагають досвіду з SQL та зв'язаних з ним стандартів, що дозволяє розробникам більш швидко розгорнути нові застосунки.

– Геопросторові запити: Деякі NoSQL бази даних, наприклад MongoDB, підтримують геопросторові запити, що дозволяє зберігати та обробляти дані, пов'язані з географічними координатами. Це може бути корисно для розробки застосунків, які використовують геоданні.

– Реплікація: Багато NoSQL баз даних дозволяють легко створювати резервні копії та реплікувати дані на інші вузли. Це дозволяє забезпечити високу доступність та надійність системи.

Нереляційні бази даних мають свої особливості та можуть бути кращим вибором для деяких типів застосунків, особливо тих, що потребують швидкої обробки великих обсягів даних, таких як соціальні мережі, IoT, аналітика даних тощо. Однак, вони також мають свої недоліки, такі як менша стандартизація та обмеження на рівні запитів.

Основні переваги MongoDB:

MongoDB є базою даних з відкритим вихідним кодом, яка надає гнучкий підхід до зберігання даних будь-якого типу та розміру. Вона проста у використанні та розгортанні, що дозволяє швидко встановлювати та налаштовувати систему.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

База даних легко масштабується, дозволяючи обробляти великі обсяги даних та підвищувати продуктивність за допомогою розширення кластера. MongoDB підтримує геопросторові запити, що може бути корисним для розробки веб-застосунків, які використовують геолокацію. MongoDB має можливість реплікації даних на кількох серверах для забезпечення надійності та доступності. Крім того, вона дозволяє шардування для розподілу даних між серверами та масштабування в горизонтальному напрямку. MongoDB також має вбудовану підтримку агрегації та операції MapReduce, що дозволяє виконувати аналітику та обробку великих обсягів даних за допомогою одного запиту.

Проектування функціональної архітектури - це процес визначення функціональних блоків та їх взаємодії в системі. Цей процес передбачає створення опису функцій, що повинні бути виконані системою, та їх подальше розбиття на менші функціональні блоки. Кожен блок містить окрему функціональну частину системи та може містити кілька компонентів.

Для проектування функціональної архітектури використовуються різні методики та підходи, які зазвичай починаються з аналізу вимог до системи. Від цього залежить визначення функцій, що повинні бути виконані системою. Після цього проводиться розбиття функцій на менші блоки та визначення їх взаємодії.

У процесі проектування функціональної архітектури можуть бути використані різні засоби моделювання, такі як UML-діаграми, блок-схеми, контекстні діаграми та інші. Ці засоби дозволяють детально описати функціональні блоки та їх взаємодію в системі.

Проектування функціональної архітектури є важливим етапом в процесі розробки програмного забезпечення, що дозволяє визначити функції, які повинна виконувати система, та їх взаємодію. Це дозволяє розробникам зосередитися на деталях реалізації окремих функціональних блоків, що полегшує процес розробки та забезпечує більшу ефективність роботи. Проектування функціональної архітектури соціальної мережі передбачає визначення функцій, які мережа повинна виконувати. Основні функції соціальної мережі :

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Створення профілю користувача: це функція, яка дозволяє користувачеві створити свій профіль зі своїми особистими даними, фотографіями та іншою інформацією, яку вони хочуть додати.

Пошук та додавання друзів: ця функція дозволяє користувачам знаходити і додавати інших користувачів як своїх друзів в мережі. Додавання друзів може включати запрошення користувачів до мережі та підтвердження запитів на дружбу.

Обмін повідомленнями та комунікація: ця функція дозволяє користувачам обмінюватися повідомленнями, фотографіями, відео та іншою інформацією зі своїми друзями та іншими користувачами мережі.

Створення та ділення контенту: ця функція дозволяє користувачам створювати та ділитися різними типами контенту, такими як фотографії, відео, статті тощо.

Новинний стрім: ця функція дозволяє користувачам отримувати оновлення від своїх друзів та підписок, такі як новини, повідомлення про події, опублікований контент тощо.

Захист персональних даних: ця функція дозволяє користувачам захищати свою особисту інформацію від доступу інших користувачів та третіх сторін. Це може включати налаштування приватності, фільтри для небажаного контенту та інші інструменти для забезпечення безпеки та конфіденційності.

Першим є процес авторизації та реєстрації, без нього неможливі будь які подальші дії користувача. Вона містить у собі такі функції:

– Аутентифікація користувача: це процес перевірки, що користувач є дійсним користувачем соціальної мережі, і отримання даних про нього, таких як ім'я, прізвище, електронна адреса тощо.

– Авторизація доступу: це процес надання користувачеві доступу до ресурсів, що належать соціальній мережі, таких як дописи, фотографії, відео, друзі тощо.

– Збереження сесії користувача: це процес збереження інформації про те, що користувач авторизувався, та забезпечення безперервного доступу до ресурсів

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						31
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

соціальної мережі після закриття браузера.

Для авторизації в соціальній мережі можна використовувати стандартні протоколи авторизації, такі як OAuth або OpenID Connect. Ці протоколи дозволяють користувачам дозволяти стороннім застосункам доступ до своїх даних без необхідності відкривати свій пароль.

Для реалізації функціональної архітектури можна використовувати різні засоби, такі як мови програмування, фреймворки, бібліотеки тощо. Наприклад, для розробки функції авторизації в соціальній мережі можна використовувати мови програмування, такі як Python, PHP, Ruby або JavaScript, та фреймворки, такі як Django, Laravel, Ruby on Rails або Express.js. Залежно від обраного засобу розробки, може бути необхідно використовувати певні бібліотеки або інші засоби для забезпечення безпеки та захисту від зловмисних атак.

На рисунку 2.1 наведена схема авторизації користувача у мережу.

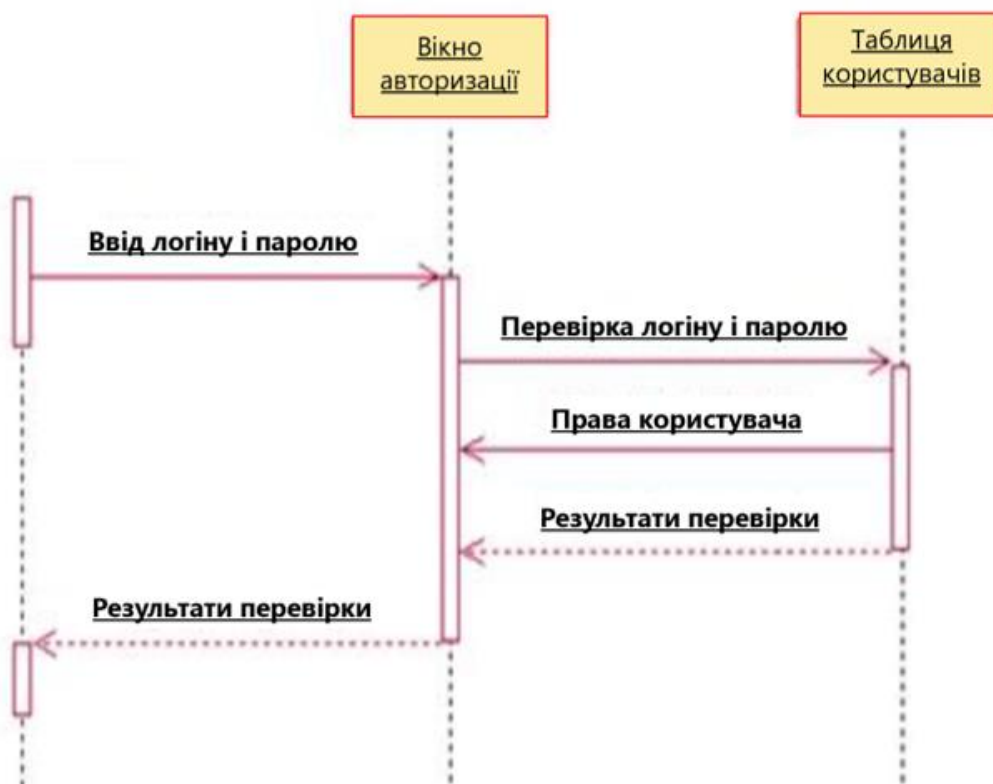


Рисунок 2.1 – Діаграма послідовності для прецедента «Авторизація»

Також важливим моментом є редагування профілю користувача. Користувач повинен мати можливість змінити персональні данні та іншу інформацію про себе у будь який момент. Для цього він повинен зайти у відповідне поле для зміни даних профілю та замінити інформацію на оновлену. Ця інформація передається на сервер, обробиться та якщо не буде виявлено помилок занесеться у базу даних. Схема зображена на рисунку 2.2.

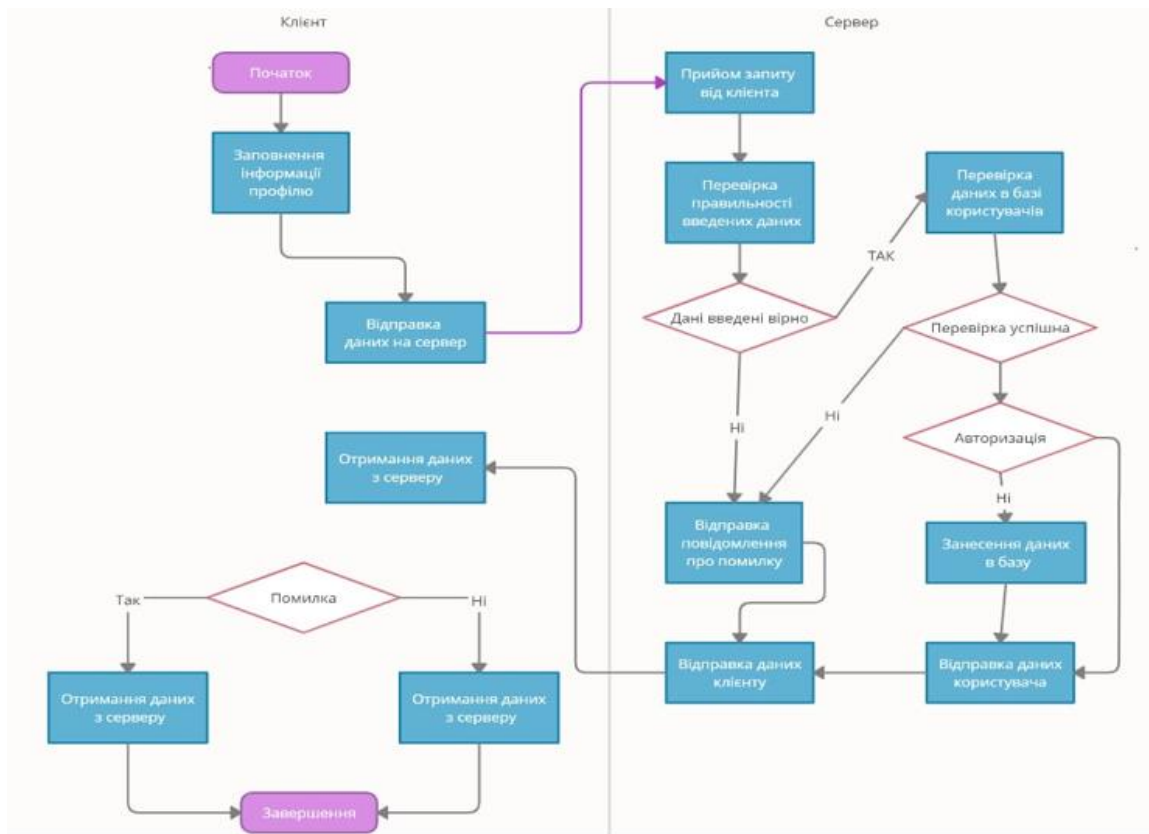


Рисунок 2.2 – Схема редагування профілю

Модуль автентифікації є першим та основним модулем соціальної мережі, який забезпечує безпеку та конфіденційність користувачів. Цей модуль складається з трьох основних функцій - реєстрації, входу та скидання паролю. Автентифікація користувачів проводиться за допомогою комбінації електронної пошти та пароля.

Під час реєстрації користувачів, модуль повинен збирати необхідну інформацію, таку як ім'я, електронну пошту та пароль, а також проводити валідацію цих даних, щоб переконатись у їхній правильності. Дані користувачів

мають бути захищені від несанкціонованого доступу та зберігатися у базі даних з використанням безпечних методів шифрування та зберігання.

Після реєстрації, користувач має можливість увійти до свого облікового запису, використовуючи свою електронну пошту та пароль. Під час процесу входу, модуль автентифікації повинен перевірити, чи існують такі дані користувача у базі даних, та перевірити правильність введених даних. При успішній перевірці, користувач має отримати доступ до свого облікового запису. Якщо користувач забув свій пароль, модуль автентифікації має мати можливість скинути його. Після заповнення форми скидання пароля, користувач має отримати електронний лист із посиланням на сторінку, де він зможе ввести новий пароль.

Модуль автентифікації має забезпечувати безпеку та конфіденційність даних користувачів, що проходять перевірку з даними, які знаходяться у базі даних. Для цього, модуль повинен використовувати безпечні методи шифрування паролів, такі як хешування з ітераціями, щоб запобігти витоків паролів у разі складнощів з безпекою системи.

Під час реєстрації, модуль повинен також перевіряти унікальність електронної пошти, щоб уникнути створення дублікатів облікових записів. При проведенні входу, модуль автентифікації повинен обробляти введені дані з урахуванням недостовірності користувачів та здійснювати захист від атак злову паролів.

Модуль "Профіль" у соціальній мережі є місцем, де користувачі можуть поділитися своєю інформацією та взаємодіяти з іншими користувачами. Цей модуль має декілька ключових елементів, що включають головну фотографію сторінки, особисту інформацію та стінку з новинами, на якій користувачі можуть ділитися своїми дописами з друзями.

На сторінці профілю користувач може змінювати свої особисті дані, включаючи головне фото, інформацію про себе, таку як місце проживання, робота, інтереси тощо. Це дає можливість користувачам показати свою особистість та створити індивідуальний профіль.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						34
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Окрім того, модуль "Профіль" надає користувачам можливість створювати нові дописи, які з'являються на їх стінці. Це дозволяє користувачам ділитися новинами, фотографіями, відео або будь-якими іншими важливими подіями чи думками зі своїми друзями. Крім того, користувачі можуть бачити реакції (лайки, коментарі) від інших користувачів на їхні дописи та взаємодіяти з ними, наприклад, відповідати на коментарі або виразити подяку.

Модуль "Профіль" є центральним місцем для представлення користувача у соціальній мережі, де він може виразити свою особистість, спілкуватися з друзями та спостерігати за активністю на своїй сторінці. Забезпечення зручного та інтуїтивно зрозумілого інтерфейсу, а також можливості контролювати конфіденційність і налаштування приватності є важливими аспектами цього модуля, щоб забезпечити задоволення та захист прав користувачів.

Модуль "Новини" у соціальній мережі є центральним місцем, де користувачі можуть переглядати та взаємодіяти з новими дописами своїх друзів. Цей модуль включає в себе список дописів, які містять інформацію про користувача, що їх створив (ім'я, прізвище та фотографію), а також зміст самого допису та час його публікації.

У модулі "Новини" користувачі можуть переглядати оновлення від своїх друзів, такі як повідомлення, фотографії, відео, посилання або будь-який інший контент, який додали користувачі, з якими вони знаходяться у списку друзів. Користувачі можуть коментувати дописи своїх друзів, виражати свої реакції (лайки, коментарі, сміх тощо) та поділитися цими дописами зі своїми власними друзями.

Крім того, модуль "Новини" може включати функції фільтрацію та сортування дописів, щоб користувачі могли переглядати контент, який їх цікавить. Наприклад, вони можуть відображати лише дописи від певних друзів, використовувати ключові слова або хештеги для пошуку певних тем, або використовувати фільтри, щоб побачити найпопулярніші або найновіші дописи.

Модуль "Новини" є важливим елементом соціальної мережі, який допомагає

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

користувачам залишатися в курсі активності їх друзів та спілкуватися з ними через коментарі та реакції. Для забезпечення приємного досвіду використання, модуль повинен бути швидким, надійним і забезпечувати зручну навігацію між дописами та взаємодію з ними.

Модуль "Друзі" у соціальній мережі є місцем, де користувачі можуть переглядати список людей, яких вони добавили у свій список друзів. У цьому модулі кожен друг відображається з особистою інформацією, такою як фотографія та інтереси, що допомагає зрозуміти ближче цю особу.

Користувачі можуть переходити на сторінку кожного друга, де можуть отримати детальнішу інформацію про нього, таку як додаткові фотографії, дописи, інформацію про роботу, освіту або інші деталі, які вони вирішили розкрити. Це дає можливість поглибленого знайомства з друзями та зміцнення зв'язків у соціальній мережі.

Крім того, модуль "Друзі" також надає можливість користувачам видаляти людей зі свого списку друзів, якщо вони більше не бажають підтримувати зв'язок з цією особою. Видалення зі списку друзів не означає блокування або видалення профілю, а просто видаляє їх зі списку приватних зв'язків.

Модуль "Друзі" є важливим елементом соціальної мережі, який допомагає користувачам підтримувати і керувати своїми соціальними зв'язками, а також забезпечує зручний доступ до особистої інформації та активності своїх друзів.

Модуль "Повідомлення" у соціальній мережі забезпечує користувачам можливість обмінюватись приватними повідомленнями один з одним. Цей модуль створений для зручного та конфіденційного спілкування між користувачами.

У модулі "Повідомлення" користувачі можуть відправляти текстові повідомлення один одному. Вони можуть створювати нові діалоги з іншими користувачами або продовжувати розмови, які вже були розпочаті. Користувачі можуть відправляти текст, фотографії, відео або інші медіафайли у своїх повідомленнях.

Модуль "Повідомлення" також надає користувачам можливість переглядати

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

історію повідомлень, включаючи всі передані повідомлення та прикріплені файли. Це дозволяє користувачам легко знаходити попередні розмови та переглядати їх в будь-який момент.

Модуль "Повідомлення" є важливим елементом соціальної мережі, який дозволяє користувачам зберігати зв'язок і спілкуватись в приватному режимі, що сприяє активному та особистому взаємодії між користувачами.

### 2.3 Проектування інтерфейсу користувача

Користувачі будуть взаємодіяти з інтерфейсом, потрібно створити простий, зручний, інтуїтивно зрозумілий інтерфейс. Сторінки повинні бути у стриманих кольорах, та не подразнювати очі. Все, що користувач побачить повинно виконувати якусь функцію, тому потрібно визначити, що ж саме користувач зможе робити. Діаграма варіантів використання наведена на рисунку 2.3.

Діаграма містить у собі перелік можливості дій користувача. Соціальна мережа повинна мати головну сторінку, з якої буде виконуватись навігація по веб-застосунку. Щоб потрапити на головну сторінку потрібно пройти етап реєстрації та авторизації. З головної сторінки користувач повинен мати доступ до своєї сторінки, повідомлень, а також вкладки друзі, новини, спільний чат.

Інтерфейс у соціальній мережі повинен відповідати наступним вимогам:

- Легкість використання: інтерфейс повинен бути простим у використанні та легким для навігації. Користувач повинен легко знаходити необхідні функції та опції.
- Зручність: інтерфейс повинен бути зручним для користування на різних пристроях, таких як комп'ютери, смартфони та планшети.
- Консистентність: інтерфейс повинен мати однаковий стиль та дизайн на всіх сторінках, щоб забезпечити користувачам стабільний досвід використання.

					КвРІПЗ.190133.19.07.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

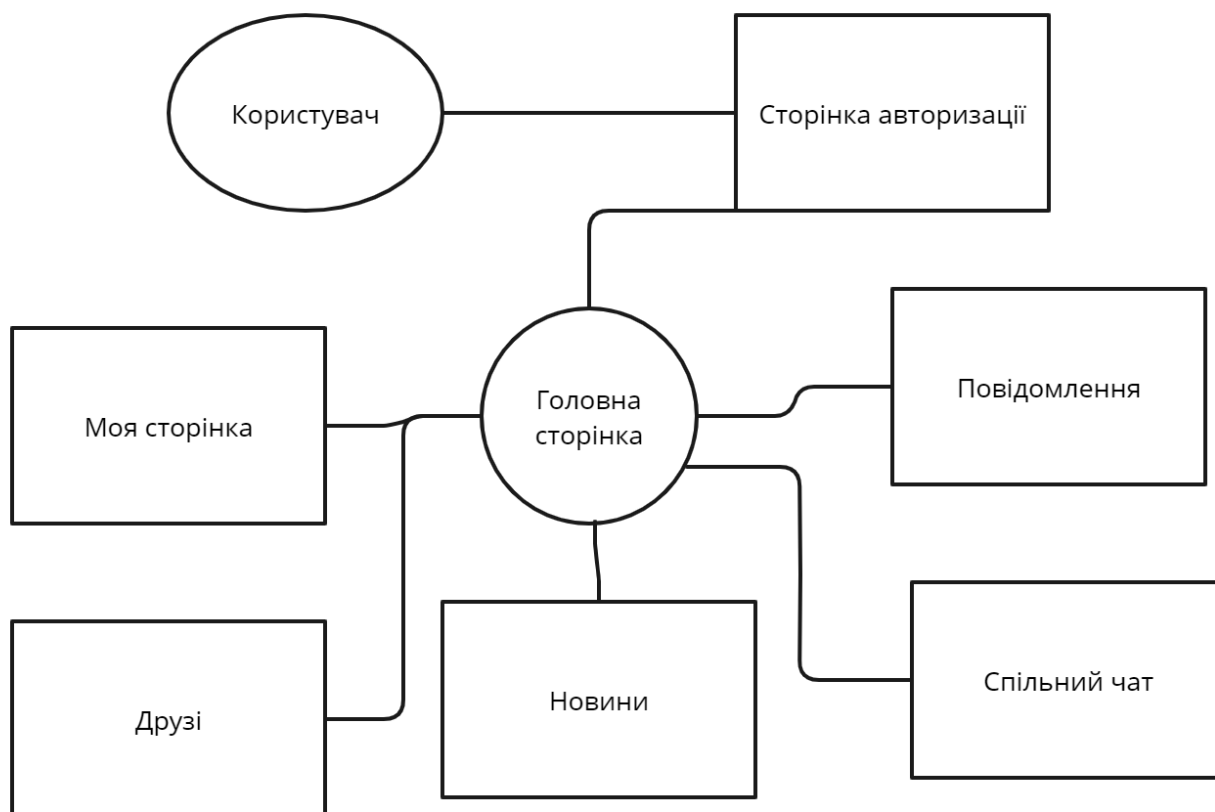


Рисунок 2.3 – Діаграма варіантів використання

- Доступність: інтерфейс повинен бути доступним для користувачів з різними потребами та здібностями.
- Безпека: інтерфейс повинен бути безпечним та захищеним від шахрайства та крадіжок даних. Користувачі повинні бути захищені від шкідливих програм, фішингових атак та злочинної діяльності.
- Функціональність: інтерфейс повинен мати всі необхідні функції, які користувачі очікують від соціальної мережі. Наприклад, можливість додавання друзів, створення та відправка повідомлень, ділитися фотографіями та відео, та інші.
- Персоналізація: інтерфейс повинен надавати користувачам можливість налаштувати свій профіль та відобразити інформацію, яка їм важлива. Користувач повинен мати можливість налаштувати свій інтерфейс відповідно до своїх вподобань та потреб.

– Соціальна взаємодія: інтерфейс повинен забезпечувати можливість спілкування та взаємодії користувачів між собою, наприклад, через коментарі, відгуки, відмітки на фотографіях, тощо.

– Швидкість та продуктивність: інтерфейс повинен бути швидким та продуктивним, щоб користувачі могли швидко отримувати доступ до інформації та взаємодіяти з іншими користувачами.

– Адаптивність: інтерфейс повинен бути адаптивним до змін у поведінці користувачів та у зміні їх потреб. Наприклад, соціальна мережа може реагувати на нові тренди та застосовувати їх у своїх функціях та дизайні.

– Розширюваність: інтерфейс повинен бути розширюваним та здатним до змін, щоб забезпечити користувачам нові можливості та функціонал у майбутньому.

– Транспарентність: інтерфейс повинен бути транспарентним щодо збору та використання даних користувачів, щоб забезпечити їхню приватність та довіру до соціальної мережі.

#### 2.4 Детальне проектування даних

Детальне проектування даних є дуже важливим при розробці соціальної мережі з наступних причин:

– Ефективність: Детальне проектування даних дозволяє оптимізувати структуру та взаємозв'язки між сутностями, що дозволяє забезпечити ефективний доступ до даних. Це важливо для соціальної мережі, оскільки вона має велику кількість користувачів та залучає значну кількість даних, таких як повідомлення, дописи, коментарі тощо. Ефективна структура даних допомагає зменшити час доступу до даних та забезпечити швидкість відгуку системи.

– Масштабовність: Детальне проектування даних дозволяє планувати масштабування соціальної мережі. Воно допомагає визначити, як дані будуть

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

зберігатися та оброблятися, коли кількість користувачів та активності в мережі зростає. Правильна архітектура даних дозволить легко масштабувати систему шляхом додавання нових серверів, розподілу навантаження та оптимізації запитів до бази даних.

– Інтеграція та розширюваність: Детальне проектування даних забезпечує гнучкість та легкість інтеграції з іншими компонентами соціальної мережі. Це дозволяє розширювати функціональність мережі, додавати нові модулі та функції без необхідності редизайну всієї системи. Наприклад, якщо потрібно додати можливість коментувати повідомлення, правильна архітектура даних дозволить легко впровадити цю функцію, розширивши відповідну сутність та зв'язки.

– Безпека та конфіденційність: Детальне проектування даних дозволяє враховувати аспекти безпеки та конфіденційності даних. Наприклад, збереження паролів у захищеному вигляді, встановлення прав доступу до конкретних даних тільки для відповідних користувачів та забезпечення конфіденційності особистої інформації користувачів. Це надзвичайно важливо для соціальних мереж, оскільки вони містять багато особистих даних користувачів, які потребують надійного захисту.

– Зручність використання: Детальне проектування даних спрощує розробку функцій та функціональності соціальної мережі. Чітка структура даних дозволяє розробникам швидко розуміти, як взаємодіють різні компоненти та як працюють запити до бази даних. Це сприяє швидкому розгортанню нових функцій та вирішенню проблем, що можуть виникати.

Основні поля схеми "Post" включають:

- text: Текст поста, який є обов'язковим полем типу String.
- isUpdated: Поле, що позначає, чи був пост оновлений. За замовчуванням, це поле має значення false, та його тип - Boolean.
- likes: Поле, яке представляє масив об'єктних ідентифікаторів (ObjectId) користувачів, які вподобають цей пост. Воно посилається на модель "User".
- owner: Поле, яке посилається на власника поста, вказуючи на його

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

об'єктний ідентифікатор (ObjectId). Воно також посилається на модель "User".

Додатково, схема має опцію {timestamps: true}, яка автоматично додає поля createdAt та updatedAt для відстеження дати створення та останнього оновлення запису. Нарешті, модель "Post" експортується для використання в інших частинах коду.

Код підключення до бази даних можна побачити у додатку Г. userSchema включає наступні поля:

- email: рядок, обов'язкове поле, яке зберігає адресу електронної пошти користувача
- username: рядок, обов'язкове поле, яке зберігає ім'я користувача
- password: рядок, обов'язкове поле, яке зберігає хеш пароля користувача
- followers: масив ідентифікаторів користувачів, на яких підписаний поточний користувач
- following: масив ідентифікаторів користувачів, на яких підписаний поточний користувач
- fullName: рядок, необов'язкове поле, яке зберігає повне ім'я користувача
- photo: рядок, необов'язкове поле, яке зберігає URL-адресу зображення профілю користувача
- status: рядок, необов'язкове поле, яке зберігає статус користувача
- aboutMe: рядок, необов'язкове поле, яке зберігає інформацію про користувача
- contacts: об'єкт, що містить наступні поля:
  - skype: рядок, необов'язкове поле, яке зберігає ім'я користувача в Skype
  - telegram: рядок, необов'язкове поле, яке зберігає ім'я користувача в Telegram
  - linkedIn: рядок, необов'язкове поле, яке зберігає URL-адресу профілю користувача в LinkedIn
  - github: рядок, необов'язкове поле, яке зберігає URL-адресу профілю користувача в GitHub

					КвРІПЗ.190133.19.07.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

– website: рядок, необов'язкове поле, яке зберігає URL-адресу веб-сайту користувача

Поля timestamps встановлені на true, щоб зберігати дату створення та оновлення запису в базі даних. В результаті моделі зберігають дату створення та оновлення кожного запису. Модель User експортується для використання в інших частинах коду.

Для підключення бази даних MongoDB до соціальної мережі потрібно встановити та налаштувати MongoDB, а також використати драйвер MongoDB для з'єднання з базою даних у своєму застосунку. Ось кроки які потрібні для коректного встановлення та роботи:

– Завантаження з офіційного сайту та встановлення MongoDB, згідно до інструкцій для вашої операційної системи.

– Запуск MongoDB у хмарній службі MongoDB Atlas. Перевірка працездатності на зручному порті.

– Встановлення пакетів Mongoose: Mongoose - це бібліотека Node.js, яка надає зручний спосіб працювати з MongoDB. Встановлення пакетів Mongoose за допомогою менеджера пакетів npm install mongoose.

– Підключення застосунку до бази даних:

Код описаний у додатку Г є основним файлом застосунку, який налаштовує підключення до бази даних MongoDB і маршрутизацію для аутентифікації, користувачів та постів. Розглянемо основні кроки, які виконуються в цьому коді:

– Рядок require('dotenv/config') дозволяє завантажити конфігурацію з файлу .env, де можна зберегти конфіденційні дані, такі як URL бази даних та URL клієнта. Створено файл .env і вказано значення для DB\_URI, PORT та CLIENT\_URL.

– Далі, налаштовується Express додаток за допомогою express(). За допомогою app.use, ми встановлюємо різні middleware для застосунку.

– app.use(fileUpload({})) дозволяє завантажувати файли на сервер.

– app.use('/static', express.static('static')) налаштовує статичну маршрутизацію для папки "static". Це дозволить вам використовувати файли з цієї

					КвРІПЗ.190133.19.07.ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

папки у вашому застосунку.

- `app.use(express.json())` дозволяє парсити JSON дані, що надходять з клієнта.
- `app.use(cookieParser())` додає middleware для обробки cookie в запитах.
- `app.use(cors({...}))` налаштовує middleware для обробки політики міжсайтових запитів (CORS). Ви повинні встановити `process.env.CLIENT_URL` у `.env` файлі, щоб дозволити запити з вашого клієнтського застосунку.
- Далі, за допомогою `app.use('/api/auth', authRouter)`, `app.use('/api/users', userRouter)`, `app.use('/api/posts', postRouter)`, встановлюються маршрути для аутентифікації, користувачів та постів. Зазначені роутери повинні бути визначені у відповідних файлів `auth.router.js`, `users.router.js` та `post.router.js`.
- Виконується підключення до бази даних MongoDB за допомогою `mongoose.connect`. Значення URL бази даних повинно бути збережене у `process.env.DB_URI` в `.env` файлі.

Якщо підключення до бази даних вдале, запускається сервер на вказаному порті (`process.env.PORT`).

## 2.5 Аналіз та вибір технологій розробки клієнтської частини

Існує багато засобів для розробки клієнтської частини соціальної мережі, але вибір конкретного залежить від технічних можливостей та вимог проекту. Ось декілька популярних засобів:

- React - це бібліотека JavaScript для створення інтерфейсу користувача. Вона дозволяє розробникам створювати динамічні веб-сторінки, які легко масштабуються. Використання React дозволить вам створити масштабований та ефективний інтерфейс соціальної мережі.
- Vue.js - це прогресивна JavaScript-бібліотека для створення інтерфейсу користувача. Вона дозволяє розробникам створювати ефективні та масштабовані

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

веб-застосунків з мінімальними зусиллями. Vue.js забезпечує багатофункціональну та легку налагодження інфраструктуру для розробки соціальної мережі.

– Angular - це фреймворк JavaScript для створення веб- застосунків. Він дозволяє розробникам швидко створювати складні веб- застосунки з великою кількістю інтерактивних елементів, які легко масштабуються. Angular забезпечує широкий набір функціональних можливостей, які зроблять вашу соціальну мережу ефективною та ергономічною.

– Bootstrap - це фреймворк HTML, CSS та JavaScript для розробки веб-застосунків та інтерфейсів. Він дозволяє розробникам створювати масштабовані та ергономічні інтерфейси, що працюють на різних пристроях та браузерах. Bootstrap забезпечує широкий набір готових компонентів та стилів, які зроблять вашу соціальну мережу привабливою та легко відтворюваною.

– Ember.js - це фреймворк JavaScript для створення веб- застосунків з високою продуктивністю та масштабованістю. Він дозволяє розробникам швидко створювати веб-застосунки, які працюють на різних пристроях та браузерах. Ember.js має багатофункціональну та легку налагодження інфраструктуру, яка забезпечує ефективну розробку соціальної мережі.

– Ionic - це фреймворк для створення мобільних застосунків з використанням HTML, CSS та JavaScript. Він дозволяє розробникам створювати нативні застосунки для різних мобільних платформ, включаючи iOS та Android. Ionic має багатофункціональну та легку налагодження інфраструктуру для розробки соціальної мережі для мобільних пристроїв.

React є однією з найпопулярніших та найбільш використовуваних бібліотек для розробки клієнтської частини веб-застосунків, в тому числі і соціальних мереж. Ось деякі переваги використання React для розробки соціальної мережі:

– Легкість використання та швидкість розробки: React має простий та логічний синтаксис, що дозволяє швидко створювати компоненти та складні інтерфейси. Крім того, наявність багатьох готових компонентів та бібліотек дозволяє зекономити час на розробці та тестуванні.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						44
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

– Висока продуктивність: React використовує віртуальний DOM, що дозволяє зменшити кількість операцій з реальним DOM та підвищити продуктивність веб- застосунку. Крім того, React має ефективну систему оновлення стану компонентів, що також сприяє покращенню продуктивності.

– Масштабованість: React дозволяє створювати складні та масштабовані веб- застосунки, які можуть працювати з великою кількістю даних та користувачів. React також має багато інструментів для тестування та підтримки коду, що сприяє забезпеченню якості розробки.

– Багатофункціональність: React підтримує багатофункціональну розробку, що дозволяє розробникам додавати до веб- застосунку такі функції, як автозаповнення, фільтрацію даних, сповіщення та багато іншого.

## 2.6 Висновки до другого розділу

React є гарним варіантом для розробки клієнтської частини соціальної мережі, це легкий та продуктивний інструмент. За допомогою React можна швидко та ефективно розробити складні інтерфейси та функціональність для соціальної мережі, що забезпечить їй конкурентну перевагу на ринку. Крім того, велика спільнота розробників та наявність багатьох готових компонентів та бібліотек дозволяє ефективно розвивати проект та зменшує ризик помилок під час розробки. Отже використавши разом React, NodeJS, MongoDB можна розробити соціальну мережу з широким функціоналом.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

#### 3.1 Програмна реалізація модулів

Під час початку використання соціальної мережі перше, що побачить користувач це модуль автентифікації, а саме модуль реєстрації. Для того щоб зареєструватися потрібно ввести електронну пошту, свій логін а також придумати пароль. Ці дані будуть надіслані у базу даних та будуть використовуватись для автентифікації користувача. Модуль проведення автентифікації та авторизації наведено в додатку В.

Код наведений в додатку представляє обробник запиту для реєстрації нового користувача у соціальній мережі. При отриманні запиту, функція отримує з тіла запиту email (електронна пошта), username (ім'я користувача) та password (пароль). Далі, за допомогою методу findOne, вона перевіряє, чи вже існує користувач з такою самою електронною поштою в базі даних.

Якщо користувач вже існує, то повертається відповідь з кодом статусу 400 і повідомленням про те, що користувач вже існує в системі.

У випадку, якщо користувач ще не існує, функція викликає метод createUser з сервісу AuthService для створення нового користувача з переданими параметрами (електронна пошта, ім'я користувача та пароль).

Після успішного створення користувача, функція встановлює куку з токеном (res.cookie('token', generateToken(user.\_id), {maxAge: 60 \* 60 \* 1000, httpOnly: true})), що дозволить зберігати токен аутентифікації на клієнтському боці протягом 60 хвилин.

Фрагмент коду наведено в додатку В. Цей код представляє обробник запиту для входу користувача у соціальну мережу. У функції login оброблюється отриманий запит. Вона отримує з тіла запиту електронну пошту та пароль.

Далі, за допомогою методу checkUser з сервісу AuthService, функція перевіряє введені дані з існуючими користувачами в базі даних. Якщо збігу не знайдено, то повертається відповідь з кодом статусу 400 та повідомленням про

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

недійсні облікові дані.

У випадку, якщо користувача знайдено та введені дані вірні, функція встановлює куку з токеном (`res.cookie('token', generateToken(user._id), {maxAge: 60 * 60 * 1000, httpOnly: true})`), що дозволяє зберігати токен аутентифікації на клієнтському боці протягом 60 хвилин.

Даний фрагмент коду показує як саме виглядає серверна частина створення нового посту. Сервер приймає дані та згідно них створює та відтворює новий пост конкретного користувача. Також цей пост може бути редаговано.

Фрагмент коду наведено в додатку В. Цей код представляє обробник запиту для оновлення допису в соціальній мережі. При отриманні запиту, функція отримує з тіла запиту `postId` (ідентифікатор допису) та `text` (новий текст допису). Далі, за допомогою методу `findById`, вона знаходить допис за його ідентифікатором в базі даних.

Після цього, функція перевіряє, чи користувач, який робить запит, є власником допису. Якщо ідентифікатор користувача, який робить запит (`req.user.toString()`), не співпадає з ідентифікатором власника допису (`post.owner.toString()`), то повертається відповідь з кодом статусу 400 і повідомленням про неможливість оновлення допису.

У випадку, коли користувач є власником допису, функція оновлює поле `text` допису з новим текстом та встановлює прапорець `isUpdated` в значення `true`. Після цього, за допомогою методу `save()`, відбувається збереження змін до бази даних.

У функції `delete` оброблюється отриманий запит. Вона отримує параметр `id` з параметрів запиту, який відповідає ідентифікатору допису, який потрібно видалити.

Далі, за допомогою методу `findById` моделі `Post`, функція знаходить допис за його ідентифікатором. Перевіряється, чи користувач, що виконує запит, є власником цього допису. Якщо ні, то повертається відповідь з кодом статусу 400 та повідомленням про неможливість видалення допису.

У випадку, якщо користувач є власником допису, функція видаляє допис за

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

допомогою методу `deleteOne()` та повертає відповідь з кодом статусу 200 та видаленим дописом.

У функції `like` оброблюється отриманий запит. Вона отримує ідентифікатор допису `postId` з тіла запиту.

Далі, за допомогою методу `findById` моделі `Post`, функція знаходить допис за його ідентифікатором. Перевіряється, чи користувач вже має вподобання на цьому дописі. Якщо так, то з використанням методу `findByIdAndUpdate`, поле `likes` допису оновлюється за допомогою оператора `$pull`, щоб видалити вподобання поточного користувача зі списку. Якщо користувач ще не вподобав допис, то поле `likes` оновлюється за допомогою оператора `$push`, щоб додати вподобання поточного користувача до списку. Після оновлення допису, функція повертає відповідь з кодом статусу 200 та оновленим дописом.

У функції `update` оброблюється отриманий запит. Вона отримує ідентифікатор користувача з об'єкта `req.user`, а також дані, які необхідно оновити, з об'єкта `req.body`.

Далі, за допомогою методу `findByIdAndUpdate` моделі `User`, функція знаходить користувача за його ідентифікатором і оновлює його поля, використовуючи дані з `req.body`. Опція `{new: true}` забезпечує повернення оновленого об'єкта користувача після оновлення.

За допомогою методу `select`, використаного після `findByIdAndUpdate`, вибираються поля, які повинні бути включені у відповідь. У цьому випадку, поля `'password'` та `'email'` виключаються з відповіді, щоб забезпечити безпеку і приватність користувача. Після успішного оновлення користувача, функція повертає відповідь з кодом статусу 200 та оновленим об'єктом користувача.

У функції `uploadPhoto` оброблюється отриманий запит. Вона отримує фотографію з об'єкта `req.files` (передбачається, що фотографія надсилається як файл у полі `photo` запиту).

Далі, за допомогою служби `FileService`, яка відповідає за збереження файлів, фотографія зберігається за допомогою методу `saveFile`. Результат цього методу,

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

шлях до збереженого файлу photoPath, отримується.

Потім, за допомогою методу findByIdAndUpdate моделі User, функція знаходить користувача за його ідентифікатором і оновлює його поле photo, встановлюючи шлях до збереженої фотографії змінної photoPath. Опція {new: true} забезпечує повернення оновленого об'єкта користувача після оновлення.

За допомогою методу select, використаного після findByIdAndUpdate, вибираються поля, які повинні бути включені у відповідь. У цьому випадку, поля 'password' та 'email' виключаються з відповіді, щоб забезпечити безпеку і приватність користувача. Після успішного завантаження фотографії та оновлення користувача, функція повертає відповідь з кодом статусу 200 та оновленим об'єктом користувача, який тепер має оновлений шлях до фотографії.

Спочатку виконується перевірка, чи користувач не намагається підписатися на самого себе. Якщо userId співпадає з ідентифікатором поточного користувача (req.user.toString()), функція повертає відповідь з кодом статусу 400 та повідомленням про неможливість підписки на себе.

Далі, за допомогою методу findById моделі User, функція знаходить користувача, на якого потрібно підписатися, за його ідентифікатором userId.

У залежності від того, чи користувач уже підписаний на цю особу, виконуються різні дії. Якщо користувач вже має userId у своєму масиві followers, то це означає, що він вже підписаний на цю особу. У цьому випадку виконуються дві операції:

- Користувач видаляє userId зі свого масиву following за допомогою методу findByIdAndUpdate.

- Користувач видаляє поточного користувача з масиву followers користувача з ідентифікатором userId за допомогою методу findByIdAndUpdate.

Якщо користувач ще не підписаний на цю особу, виконуються такі дії:

- Користувач додає userId до свого масиву following за допомогою методу findByIdAndUpdate.

- Користувач додає поточного користувача до масиву followers

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

користувача з ідентифікатором `userId` за допомогою методу `findByIdAndUpdate`.

Компонент `Login` перевіряє, чи користувач вже авторизований (значення `isAuth` у стані). Якщо користувач вже авторизований, компонент перенаправляє його на сторінку профілю за допомогою компоненти `Navigate` з `react-router-dom`.

У випадку, якщо користувач ще не авторизований, відображається компонент `LoginForm`, який приймає функцію `login` як пропс. Ця функція викликає диспетчер `Redux` для виконання дії `login`, яка відправляє дані форми логіну на сервер для перевірки. Компонент також містить посилання на сторінку реєстрації за допомогою компоненти `NavLink` з `react-router-dom`. Якщо користувач ще не має облікового запису, він може натиснути на це посилання та перейти на сторінку реєстрації.

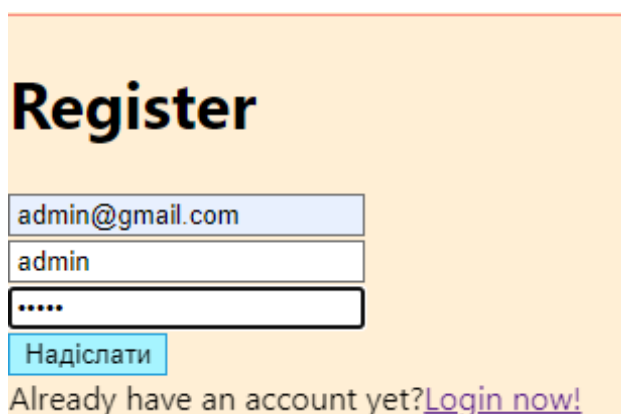
Код можна побачити у додатку Д. Компонент отримує інформацію про профіль користувача зі сховища `Redux` за допомогою хука `useSelector`. Він також відправляє дію `getUserProfile`, щоб отримати інформацію про профіль користувача при завантаженні компонента. Параметр `id` в URL використовується для визначення того, чи потрібно відобразити профіль поточного користувача чи профіль іншого користувача. Якщо `id` присутній в URL, він відправляє дію `getUserProfile` з параметром `id` для отримання інформації про профіль іншого користувача. Якщо `id` відсутній в URL, він відправляє дію `getUserProfile` з параметром `authUserId` для отримання інформації про профіль поточного користувача.

Якщо користувач не має аутентифікації і в URL відсутній параметр `id`, компонент перенаправляє користувача на сторінку входу в систему за допомогою компонента `Navigate` з пакету `react-router-dom`. Якщо інформація про профіль користувача ще не завантажена, компонент відображає повідомлення "loading". При завантаженні, він відображає компоненти `ProfileInfo` та `MyPosts`, передаючи значення `isOwner` як `true`, якщо в URL відсутній параметр `id`, що вказує на те, що профіль належить поточному користувачу.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.2 Керівництво користувача

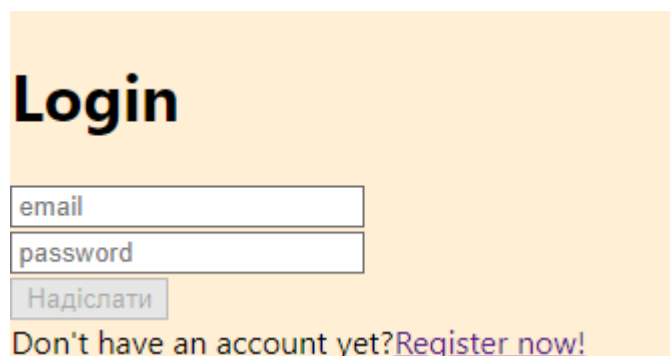
Для початку роботи з соціальною мережею користувач зареєструватися, для цього йому потрібно відкрити сторінку соціальної мережі та знайти кнопку, що відповідає за реєстрацію нового користувача. Перейшовши на сторінку реєстрації користувачу потрібно ввести наступні дані: валідна електронна адреса, логін, та пароль. Після введення та перевірки даних потрібно натиснути на кнопку «Надіслати». Приклад заповнення даних для реєстрації нового користувача можна побачити на рисунку 3 .1.



The image shows a registration form with a light orange background. At the top, the word "Register" is written in a large, bold, black font. Below it are three input fields: the first contains "admin@gmail.com", the second contains "admin", and the third contains six dots representing a password. A blue button with the text "Надіслати" (Send) is positioned below the password field. At the bottom of the form, there is a link that says "Already have an account yet? [Login now!](#)".

Рисунок 3.1 Форма реєстрації соціальної мережі

Далі користувач повинен авторизуватися у системі. Для цього йому потрібно ввести електронну адресу в поле «email» та пароль в поле«password». Якщо користувач ввів дані вказані при реєстрації, то після їх перевірки він попаде на сторінку свого профілю.



The image shows a login form with a light orange background. At the top, the word "Login" is written in a large, bold, black font. Below it are two input fields: the first is labeled "email" and the second is labeled "password". A blue button with the text "Надіслати" (Send) is positioned below the password field. At the bottom of the form, there is a link that says "Don't have an account yet? [Register now!](#)".

Рисунок 3.2 Форма авторизації соціальної мережі

Після успішної авторизації користувач може ввести інформацію про себе та хобі. Щоб завантажити аватар потрібно натиснути на кнопку «Вибрати файл» та з свого девайсу завантажити фотозображення на сервер. Щоб змінити іншу інформацію потрібно натиснути на кнопку «EditProfile» та змінити інформацію про себе. Після потрібно натиснути на кнопку «Надіслати» для збереження введених даних.



Рисунок 3.3 Форма профілю соціальної мережі

Також на цій сторінці користувач має змогу створити новий пост, редагувати попередні пости та дивитися на кількість реакцій щодо його посту. Щоб створити новий пост користувачу потрібно знайти поле для введення інформації та заповнити його контентом. Для того щоб опублікувати пост потрібно натиснути кнопку «Надіслати».

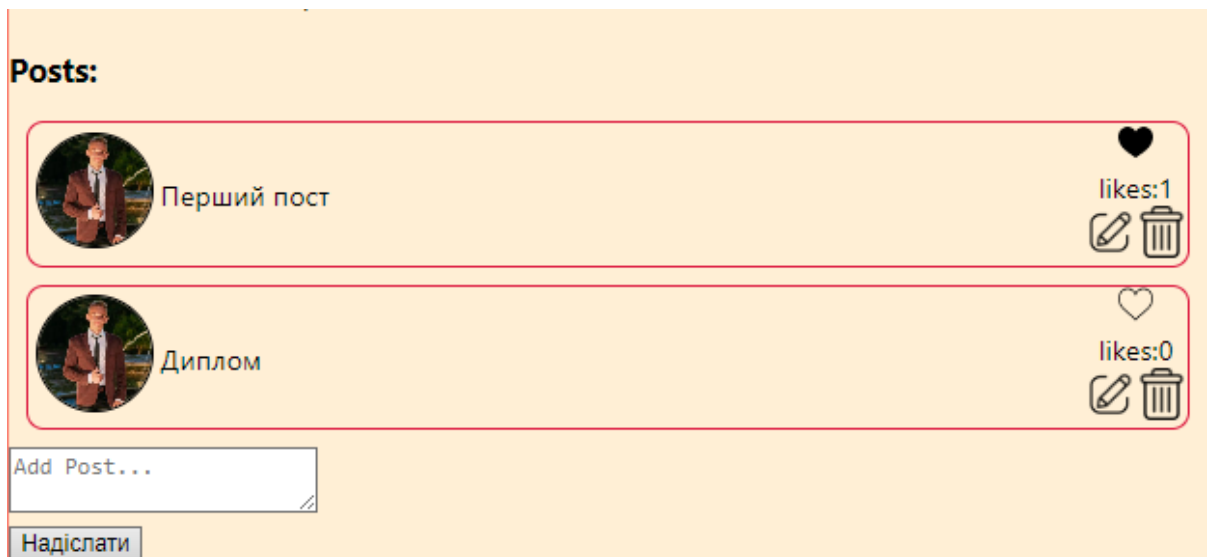


Рисунок 3.4 Форма профілю з прикладом постів соціальної мережі

Далі користувач може перейти на інші сторінки соціальної мережі. Це може бути сторінка з користувачами на яких підписаний авторизований користувач, сторінка з приватними повідомленнями та сторінка новин. Також можна переходити на сторінки інших користувачів.

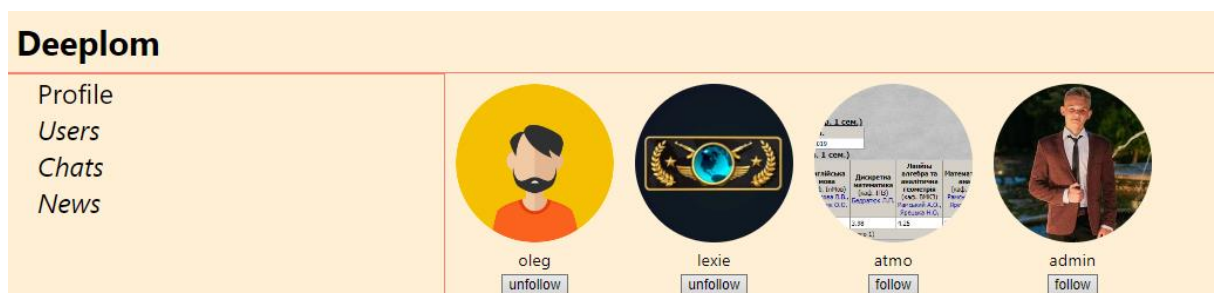


Рисунок 3.5 Форма профілю з прикладом постів соціальної мережі

Користувач має змогу підписатись на іншого натиснувши кнопку «follow» та відписатись відповідно натиснувши кнопку «unfollow». Також натиснувши на фото іншого користувача можна перейти на його профіль.

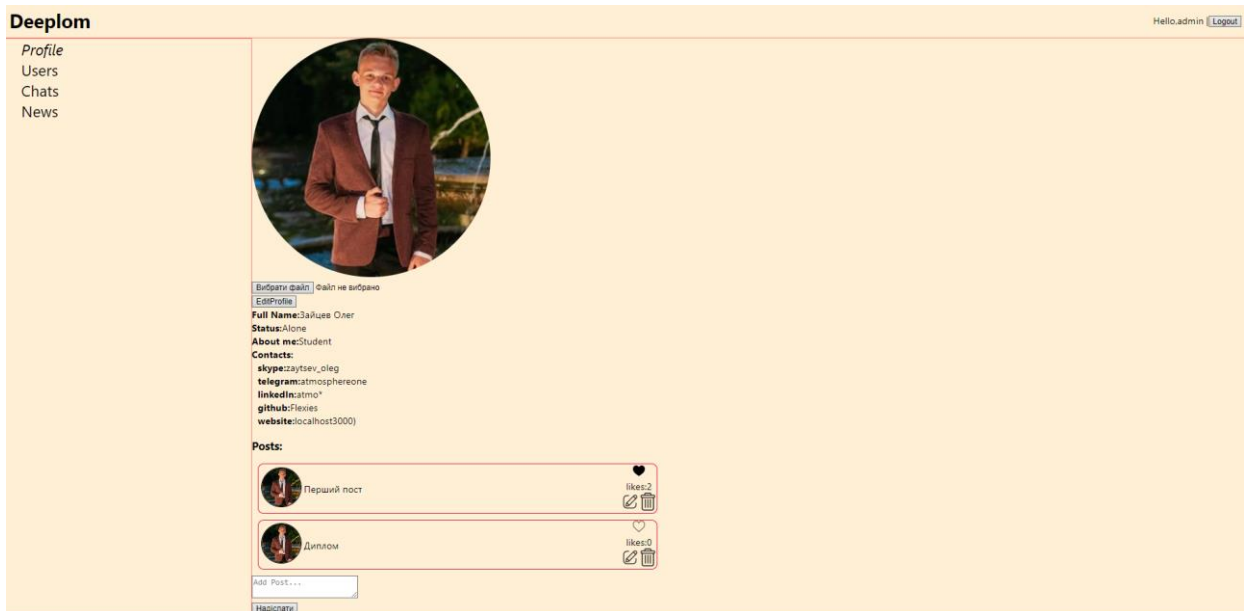


Рисунок 3.6 Форма профілю іншого користувача соціальної мережі

Також можна обмінюватись повідомленнями з користувачами перейшовши у чати.

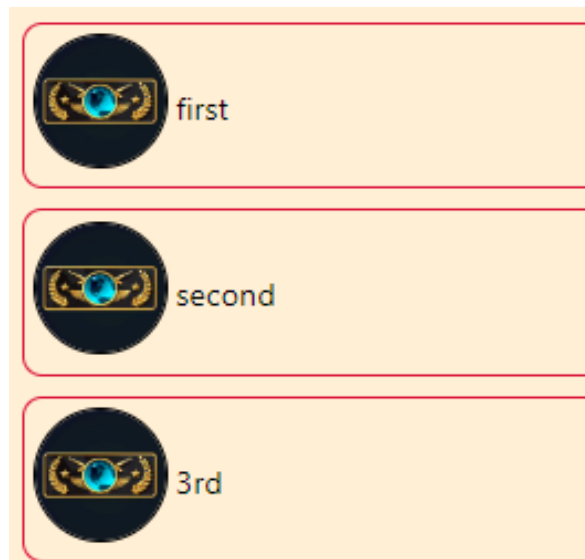


Рисунок 3.7 Форма зразку повідомлень у соціальній мережі

### 3.3 Вимоги до технічних та програмних засобів

Вимоги до технічних та програмних засобів для коректної роботи соціальної мережі:

- Операційна система: підтримувана операційна система, така як Windows 7 або новіша, macOS 10.12 або новіша або Linux з оновленнями до останньої версії.
- Процесор: мінімум двоядерний процесор з частотою не менше 2 ГГц.
- Оперативна пам'ять: рекомендовано мінімум 4 ГБ оперативної пам'яті для оптимальної продуктивності.
- Відеокарта: підтримка графічного прискорення і відеокарта з пам'яттю не менше 512 МБ.
- Жорсткий диск: вільне місце на жорсткому диску для зберігання веб-браузера та тимчасових файлів.
- Веб-браузер: оновлений веб-браузер, такий як Google Chrome, Mozilla Firefox, Safari або Microsoft Edge. Рекомендується використовувати останню доступну версію браузера для найкращого досвіду.
- Інтернет-з'єднання: стабільне інтернет-з'єднання з мінімальною швидкістю для завантаження сторінок і мультимедійного контенту.

### 3.4 Вибір та обґрунтування методів тестування веб-застосунку

Під час тестування соціальної мережі важливо враховувати різні методи тестування, які охоплюють різні аспекти системи. Ось кілька методів тестування, які найкраще підходять для соціальної мережі:

- Функціональне тестування: це включає тестування функціональності соціальної мережі, такі як реєстрація користувачів, вхід у систему, створення профілів, публікації, коментування, оцінювання, обмін повідомленнями. Воно перевіряє, чи працюють всі функції та взаємодії, як задумано.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

– Тестування зручності використання: це спрямовано на оцінку зручності використання соціальної мережі. Тестувальники можуть виконувати завдання, такі як створення облікового запису, навігація по інтерфейсу та доступ до різних функцій, щоб оцінити, наскільки інтуїтивним і зручним є платформа.

– Тестування продуктивності: це тестування продуктивності та масштабованості соціальної мережі, особливо в періоди пікового навантаження. Воно включає моделювання великого обсягу користувачів, взаємодій та даних, щоб виміряти час реакції системи, використання ресурсів та масштабованість.

Функціональне тестування є одним з ключових методів тестування соціальних мереж і включає перевірку функціональності різних аспектів платформи. Основна мета функціонального тестування полягає в перевірці, чи працюють всі функції та взаємодії, як задумано, і чи задовольняють вони вимоги користувачів.

Основні аспекти, які підлягають функціональному тестуванню соціальних мереж, включають:

– Реєстрація користувачів: перевірка правильності процесу реєстрації, включаючи валідацію введених даних, створення унікальних облікових записів користувачів та збереження їх у базі даних.

– Вхід у систему: перевірка коректності авторизації користувачів з використанням правильних облікових даних, перевірка випадків неправильного входу у систему, а також відновлення паролю.

– Створення профілів: перевірка можливості створення та налаштування профілів користувачів, додавання та оновлення особистої інформації, встановлення фотографій профілю.

– Публікації: перевірка можливості створення та редагування публікацій, включаючи додавання тексту, зображень, відео та інших медіа-елементів, налаштування приватності публікацій.

– Коментування та оцінювання: перевірка можливості користувачів коментувати публікації і ставити оцінки, перевірка правильності відображення

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

коментарів та рейтингів.

– Обмін повідомленнями: перевірка функцій обміну повідомленнями між користувачами, включаючи відправлення, отримання, відповіді та видалення повідомлень.

Ці аспекти функціонального тестування допомагають забезпечити, що соціальна мережа працює належним чином, відповідає потребам користувачів і забезпечує зручну та надійну взаємодію між користувачами.

Тестування зручності використання спрямоване на оцінку того, наскільки легко та зручно користуватися соціальною мережею з точки зору кінцевого користувача. Цей вид тестування допомагає виявити проблеми з інтерфейсом користувача, навігацією та функціональністю, які можуть ускладнювати або призводити до незадоволення користувачів.

У процесі тестування зручності використання тестувальники виконують різні завдання та сценарії, щоб оцінити зручність та інтуїтивність взаємодії з соціальною мережею. Основні аспекти, які перевіряються, включають:

– Створення облікового запису: оцінка процесу реєстрації, введення обов'язкових даних, складність паролю, наявність підказок та пояснень.

– Навігація та інтерфейс: перевірка зручності навігації по різних розділах та функціях соціальної мережі, наявність логічної структури меню та кнопок, доступ до основних функцій з легкістю.

– Взаємодія з контентом: перевірка простоти коментування, відповіді на повідомлення, оцінювання публікацій, додавання та редагування контенту (фотографій, відео, тощо).

– Пошук та фільтрація: перевірка зручності використання пошукових функцій, можливості фільтрування контенту за різними критеріями (наприклад, дата, місцезнаходження, тема).

Ці аспекти допомагають забезпечити, що соціальна мережа є зручною для використання користувачами різного рівня технічної експертизи і допомагає забезпечити задоволення від взаємодії з платформою.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						57
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Тестування продуктивності полягає у випробуванні ефективності та масштабованості соціальної мережі, особливо під час пікових навантажень. Його метою є моделювання великого обсягу користувачів, взаємодій та даних для вимірювання часу реакції системи, використання ресурсів та здатності мережі масштабуватися.

Під час тестування продуктивності важливо враховувати наступні аспекти:

– Навантаження тесту: створення сценаріїв тесту, які моделюють реальні умови використання соціальної мережі. Це може включати симуляцію одночасної активності багатьох користувачів, публікацію великої кількості повідомлень або завантаження великих файлів.

– Вимірювання продуктивності: збір метрик, таких як час відповіді сервера, час завантаження сторінок, швидкість виконання запитів до бази даних. Це дозволяє виявити можливі проблеми з продуктивністю та ідентифікувати причини затримок або перевантаження.

– Масштабованість: перевірка здатності соціальної мережі працювати ефективно при зростанні обсягу користувачів та даних. Це включає випробування системи при різних рівнях навантаження для виявлення можливих меж її масштабованості.

– Оптимізація та масштабування: ідентифікація проблемних частин системи та виявлення можливих шляхів оптимізації для поліпшення продуктивності. Це може включати використання кешування, оптимізацію запитів до бази даних та розподілення навантаження.

Тестування продуктивності допомагає забезпечити, що соціальна мережа може ефективно працювати в умовах високого навантаження та масштабуватися для забезпечення зручного та надійного використання користувачами.

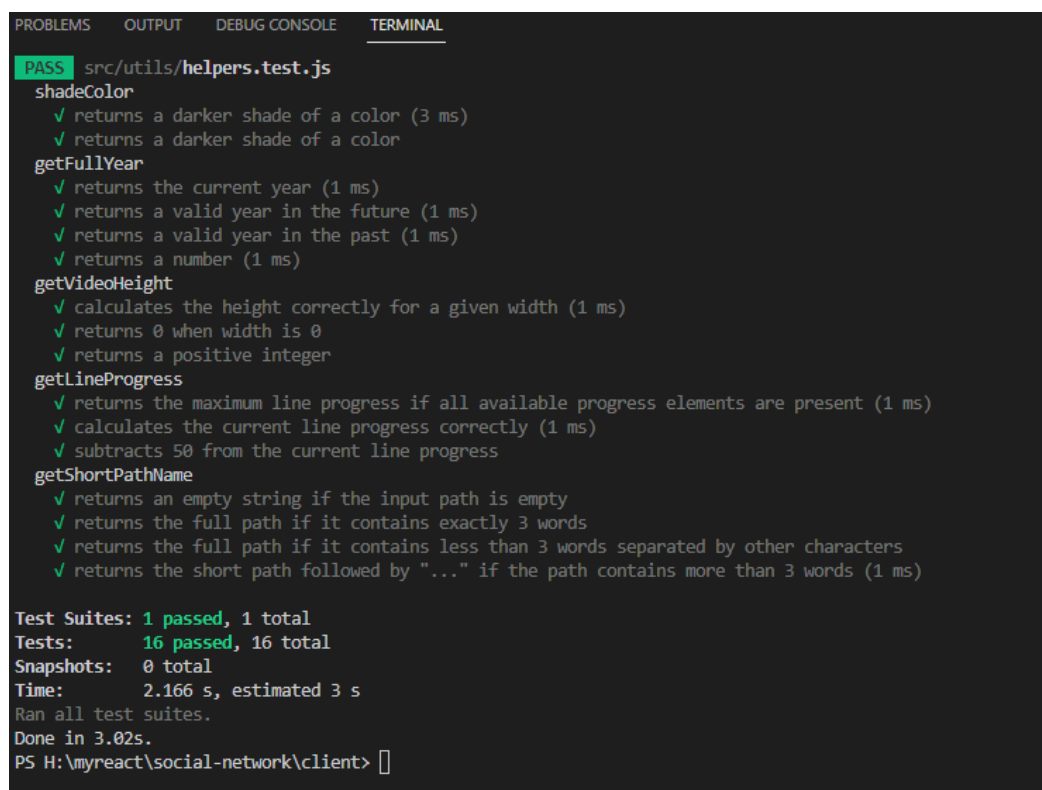
					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						58
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## Перевірка на помилки за допомогою юніт тестів

Для проведення юніт тестів було використано фреймворк jest, для цього було виконано наступні кроки:

- Встановлено фреймворк для тестування. Виконано `npm install jest` для встановлення фреймворку Jest.
- Створено файли з тестами для компонентів, функцій і класів, які потрібно перевірити. У форматі `.test.js`.
- В цьому файлі визначено тести за допомогою функції `test()` або `it()`. У цих тестах перевірено поведінку окремих функцій або методів шляхом виклику їх та перевірки результатів за допомогою різних методів асертів (наприклад, `expect()`).
- Проведено тести, виконавши команду `npm test` Фреймворк автоматично виявив та виконав всі тести вказаного файлу.
- Перевірено результати тестів.

Приклад тестування наведено у додатку Г.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PASS src/utils/helpers.test.js
shadeColor
  ✓ returns a darker shade of a color (3 ms)
  ✓ returns a darker shade of a color
getFullYear
  ✓ returns the current year (1 ms)
  ✓ returns a valid year in the future (1 ms)
  ✓ returns a valid year in the past (1 ms)
  ✓ returns a number (1 ms)
getVideoHeight
  ✓ calculates the height correctly for a given width (1 ms)
  ✓ returns 0 when width is 0
  ✓ returns a positive integer
getLineProgress
  ✓ returns the maximum line progress if all available progress elements are present (1 ms)
  ✓ calculates the current line progress correctly (1 ms)
  ✓ subtracts 50 from the current line progress
getShortPathName
  ✓ returns an empty string if the input path is empty
  ✓ returns the full path if it contains exactly 3 words
  ✓ returns the full path if it contains less than 3 words separated by other characters
  ✓ returns the short path followed by "..." if the path contains more than 3 words (1 ms)

Test Suites: 1 passed, 1 total
Tests:       16 passed, 16 total
Snapshots:  0 total
Time:        2.166 s, estimated 3 s
Ran all test suites.
Done in 3.02s.
PS H:\myreact\social-network\client> 
```

Рисунок 3.8 Результат роботи unit-тестів

Набір тестів охоплює функції `shadeColor`, `getFullYear`, `getVideoHeight`, `getLineProgress` та `getShortPathName`. Він включає кілька тестових випадків для кожної функції, які перевіряють різні сценарії та враховують граничні випадки. У тестах перевіряються очікувані значення результатів, порівняння тверджень та перевірки типів даних.

Крім того, тестові випадки для `getLineProgress` та `getShortPathName` містять описи, що пояснюють, що саме перевіряється в кожному тесті. Загалом, тестовий набір забезпечує хороше охоплення для функцій і допомагає переконатися в їх правильності та надійності. Результати unit-тестів, побачити їх можна на рисунку 3.8.

### 3.6 Валідація та верифікація веб-застосунку

Валідація та верифікація соціальної мережі є важливими кроками для забезпечення якості та надійності її функціонування. Основні причини проведення валідації та верифікації соціальної мережі такі:

- Переконатися в правильності реалізації функцій: Валідація допомагає перевірити, чи працюють основні функції соціальної мережі належним чином та не містять помилок. Це включає реєстрацію, вхід в систему, взаємодію з користувачами, публікацію контенту та інші ключові функції.

- Забезпечити безпеку та конфіденційність даних: Верифікація спрямована на перевірку безпеки соціальної мережі та захисту персональних даних користувачів. Це включає перевірку на наявність вразливостей, захист від несанкціонованого доступу та збереження конфіденційної інформації.

- Взаємодія з іншими системами: Верифікація допомагає переконатися, що соціальна мережа належним чином взаємодіє з іншими системами, такими як системи авторизації, платіжні системи, електронна пошта тощо. Це важливо для забезпечення безпроблемної роботи та інтеграції з іншими сервісами.

Перевірка на валідацію була проведена наступним чином.

					<i>KePIПЗ.190133.19.07.ПЗ</i>	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

– Перевірка валідації була розпочата з перевірки основних функцій соціальної мережі, таких як реєстрація, вхід в систему, створення профілю, додавання друзів, публікація повідомлень та інших. Я переконався, що ці функції працюють належним чином та не містять помилок.

– Далі, я перевіряв правильність відображення даних користувачів та їх безпечно зберігання. Я переконався, що конфіденційна інформація користувачів, така як паролі та персональні дані, захищена від несанкціонованого доступу.

– Я також перевіряв коректність роботи системи повідомлень, коментарів та інших способів взаємодії користувачів. Переконався, що повідомлення відправляються та отримуються коректно, а коментарі та інші взаємодії відображаються на відповідних сторінках користувачів.

Перевірка на верифікацію була проведена наступним чином.

– Для початку, я перевіряв, чи соціальна мережа відповідає вимогам безпеки та захисту даних. Я переконався, що система захищена від вразливостей, таких як атаки типу кроссайтовий скриптинг, витоки даних.

– Щоб переконатися, що соціальна мережа може витримати велику кількість користувачів та навантаження на сервери, я провів тестування навантаження та стрес-тестування. Це допомогло виявити можливі проблеми та збої при великому навантаженні.

– Я також перевіряв взаємодію соціальної мережі з іншими системами, такими як системи авторизації. Переконався, що інтеграція з цими системами працює належним чином та не виникає проблем під час взаємодії.

Також було проведено тестування на різних пристроях та платформах. Я переконався, що соціальна мережа працює належним чином на різних браузерях, операційних системах та пристроях.

Після проведення тестування соціальної мережі в цілому, ми зробили деякі спостереження та прийшли до наступних висновків:

– Основні функції соціальної мережі, такі як реєстрація, вхід в систему, створення профілю, додавання друзів, публікація повідомлень, працюють

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						61
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

належним чином і не виявлено значних помилок чи неполадок.

– Дані користувачів відображаються правильно і зберігаються безпечно. Застосовані заходи для захисту конфіденційної інформації, такої як паролі та персональні дані, виявилися ефективними та не спостерігається несанкціонованого доступу до цих даних.

– Система повідомлень, коментарів та інших способів взаємодії користувачів працює коректно. Повідомлення успішно відправляються та отримуються, а коментарі та інші взаємодії відображаються на відповідних сторінках користувачів.

– Соціальна мережа відповідає вимогам безпеки та захисту даних. Застосовані заходи для запобігання вразливостей, таких як атаки XSS та SQL-ін'єкції, показалися ефективними. Немає виявлених вразливостей чи витоків даних.

– Тестування навантаження та стрес-тестування показали, що соціальна мережа може витримати велику кількість користувачів та навантаження на сервери без помітних збоїв чи перебоїв.

– Взаємодія соціальної мережі з іншими системами, такими як системи авторизації та іншими, працює належним чином. Інтеграція з цими системами була успішною та функціонує без помилок.

Було проведено тестування системи соціальної мережі з використанням різних наборів даних. Усі тести були успішно виконані, що підтверджує коректну роботу програмної системи.

Також було проведено оптимізаційне тестування системи, яке включало вимірювання навантаження на базу даних та окремі її компоненти, зокрема таблиці. Для цього використовувався планувальник запитів, який дозволив виявити та оптимізувати неоптимальні запити з метою покращення продуктивності програмної системи.

Було проведено тестування соціальної мережі різними способами. Приклади декількох тест-кейсів для тестування функціональності соціальної мережі та їх опис буде подано нижче. Результати тестування наведені у таблиці 3.1.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						62
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Таблиця 3.1 – Основні тест-кейси

Підмодуль застосунку	Вихідні дані	Очікуваний результат
Форма реєстрації	<ol style="list-style-type: none"> <li>1. Натиснути на кнопку «register».</li> <li>2. Заповнити поля електронна пошта, логін та пароль.</li> <li>3. Клацнути на кнопку «надіслати».</li> </ol>	<ol style="list-style-type: none"> <li>1. З'являється форма реєстрації.</li> <li>2. Відбувається валідація даних.</li> <li>3. Проводиться процес автентифікації та авторизації користувача.</li> </ol>
Форма авторизації	<ol style="list-style-type: none"> <li>1. Натиснути на кнопку «login».</li> <li>2. Заповнити поля логін та пароль.</li> <li>3. Клацнути на кнопку «надіслати».</li> </ol>	<ol style="list-style-type: none"> <li>1. З'являється форма авторизації.</li> <li>2. Відбувається валідація даних.</li> <li>3. Проводиться процес реєстрації користувача.</li> </ol>
Форма сторінки профілю	<ol style="list-style-type: none"> <li>1. Натиснути на кнопку «Вибрати файл» та обрати потрібне фото.</li> <li>2. Натиснути на кнопку «EditProfile».</li> <li>3. Заповнити поля, що потребують заміни інформації.</li> <li>4. Клацнути на кнопку «Надіслати».</li> <li>5. Натиснути на поле для вводу даних «AddPost...» та ввести інформацію щодо нового допису.</li> <li>6. Клацнути на кнопку «надіслати».</li> </ol>	<ol style="list-style-type: none"> <li>1. Відбувається заміна аватару користувача.</li> <li>2. Відкривається можливість редагування персональних даних.</li> <li>3. Відбувається процес заміни даних.</li> <li>4. Збереження даних на сервері та оновлення інформації у профілі.</li> <li>5. Відкривається поле для вводу тексту.</li> <li>6. Пост зберігається на сервері та публікується на сторінці.</li> </ol>
Форма користувачів	<ol style="list-style-type: none"> <li>1. Клацнути на кнопку «follow».</li> <li>2. Клацнути на кнопку «unfollow».</li> <li>3. Клацнути на аватар користувача.</li> </ol>	<ol style="list-style-type: none"> <li>1. Відбувається процес підписки на сторінку користувача, вносяться зміни інформації на сервері.</li> <li>2. Відбувається процес відписки від сторінку користувача, вносяться зміни інформації на сервері.</li> </ol>

Змн.	Арк.	№ докум.	Підпис	Дата

КвРІПЗ.190133.19.07.ПЗ

Арк.

63

		3. Відбувається процес переходу на профіль користувача.
Форма профілю іншого користувача.	1. Натиснути на сердечко, щоб вподобати пост користувача. 2. Натиснути на пост, щоб відкрити вікно діалогу.	1. Змінюється кількість вподобайок на дописі та сервері. 2. Переход у вікно діалогів.
Форма діалогу з користувачем	1. Натиснути на поле для вводу даних «AddMessage...» та ввести текст повідомлення. 2. Клацнути на кнопку «надіслати».	1. Формування нового повідомлення. 2. Відправка нового повідомлення на сервер та до користувача.
Форма новин	1. Натиснути на сердечко, щоб вподобати пост користувача. 2. Натиснути на поле для вводу даних «AddComment...» та ввести коментар.	1. Змінюється кількість вподобайок на дописі та сервері. 2. Відправка нового коментаря на сервер та до посту.

Кінець таблиці 3.1

### 3.7 Висновки до третього розділу

У результаті тестування програмної системи соціальної мережі було виявлено, що всі тести успішно пройшли. Виявлені дефекти були успішно виправлені, а базова функціональність додатку була реалізована відповідно до вимог до програмного забезпечення. Більшість функцій є повністю працездатними, забезпечуючи коректну роботу системи соціальної мережі.

					КвРІПЗ.190133.19.07.ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

Під час виконання кваліфікаційної нам вдалося досягти поставленої мети, яка полягала в розробці системи керування контентом для веб-сайту соціальної мережі.

У першому розділі нашої роботи проведено дослідження предметної області, визначено функціональні та нефункціональні вимоги до системи керування контентом. Також було сформульовано технічне завдання та постановку задачі. Ми проаналізували наявні рішення у сфері соціальних мереж, оцінили їх позитивні та негативні аспекти.

У другому розділі ми провели проектування соціальної мережі, включаючи розробку архітектури, структури, бази даних та інтерфейсу користувача. Ми також здійснили аналіз і вибрали технології та методи реалізації, забезпечивши оптимальність та ефективність системи.

У третьому розділі ми здійснили програмну реалізацію соціальної мережі. Ми представили структуру та функціональне призначення модулів програмного засобу, розробили та впровадили бази даних, а також спроектували зручний та інтуїтивно зрозумілий інтерфейс.

У четвертому розділі ми провели тестування створеного програмного засобу в широкому діапазоні. Були виконані різноманітні тест-кейси, що дозволило перевірити коректність роботи системи та виявити та усунути можливі дефекти.

В результаті виконання цієї кваліфікаційної роботи було досягнуто мети, система керування контентом для веб-сайту соціальної мережі була успішно розроблена та протестована.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Л. П. Бедратюк. Дипломний проєкт: методичні вказівки щодо його виконання для студентів спеціальності 121 «Інженерія програмного забезпечення» / Л. П. Бедратюк, Г. І. Радельчук, Ю. В. Форкун, О. М. Яшина. Хмельницький: ХНУ, 2020. – 77 с.
2. Lon Safko, David K. Brake. The Social Media Bible: Tactics, Tools, and Strategies for Business Success . New Jersey : John Wiley & Sons, 2012. 640 с.
3. Maksim Tsvetovat, Alexander Kouznetsov. Social Network Analysis
4. Startups: Finding Connections on the Social Web. California : O'Reilly Media, 2011. 448 с.
5. Jan Zimmerman, Deborah Ng. Social Media Marketing All-in-One For Dummies. New Jersey : Wiley, 2021. 800 с.
6. Guy Kawasaki, Peg Fitzpatrick. The Art of Social Media: Power Tips for Power Users. Lonodon : Penguin Books, 2014. 208 с.
7. Christian Crumlish, Erin Malone. Designing Social Interfaces: Principles, Patterns, and Practices for Improving the User Experience. California : O'Reilly Media, 2009. 592 с.
8. Ірина Кулініч, Олена Іванченко. Соціальні медіа: стратегії та практики. Київ : Наш Формат, 2019. 256 с.
9. Stoyan Stefanov. React: Up & Running: Building Web Applications. California : O'Reilly Media, 2016. 222 с.
10. Mark Thomas. React in Action. New York : Manning, 2018. 360 с.
11. Anthony Accomazzo, Nate Murray, Ari Lerner. Fullstack React: The Complete Guide to ReactJS and Friends. New York : Fullstack.io, 2017, 836 с.
12. Carlos Santana Roldán. React Design Patterns and Best Practices. Birmingham : Packt, 2019. 318 с.
13. Marc Wandschneider. Learning Node.js. Boston : Addison-Wesley Professional, 2016. 320 с.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

14. David Herron. Node.js Web Development. Mumbai : Packt Publishing, 2017, 376 с.
15. Pedro Teixeira. Professional Node.js: Building JavaScript-Based Scalable Software. Birmingham : Wrox, 2012. 408 с.
16. Mario Casciaro. Node.js Design Patterns. Mumbai : Packt Publishing, 2020. 664 с.
17. Sandro Pasquali. Mastering Node.js. Mumbai : Packt Publishing, 2013. 346 с.
18. Jim Wilson. Node.js 8 the Right Way: Practical, Server-Side JavaScript That Scales. Raleigh : Pragmatic bookshelf, 2018. 336 с.
19. Patrick Mulder. Node.js for Embedded Systems: Using Web Technologies to Build Connected Devices. California : O'Reilly Media, 2016. 266 с.
20. Ethan Brown. Web Development with Node and Express: Leveraging the JavaScript Stack. California : O'Reilly Media, 2014. 332 с.
21. Chris Richardson. Microservices Patterns: With examples in Java. New York : Manning Publications, 2018. 520 с.
22. Developer Documentation and APIs [Електронний ресурс] // Facebook – Режим доступу до ресурсу: <https://developers.facebook.com/> (дата звернення – 13.04.2023). – Назва з екрану.
23. Twitter Developer Platform [Електронний ресурс] // Twitter – Режим доступу до ресурсу: <https://developer.twitter.com//> (дата звернення – 15.04.2023). – Назва з екрану.
24. LinkedIn Developer Network [Електронний ресурс] // LinkedIn – Режим доступу до ресурсу: <https://developer.linkedin.com//> (дата звернення – 19.04.2023). – Назва з екрану.
25. Instagram Developer Documentation [Електронний ресурс] // Instagram – Режим доступу до ресурсу: <https://developers.facebook.com/docs/instagram/> (дата звернення – 23.04.2023). – Назва з екрану.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						67
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

26. Node.js [Електронний ресурс] // Node.js– Режим доступу до ресурсу: <https://nodejs.org/dist/latest-v20.x/docs/api/> (дата звернення – 26.04.2023). – Назва з екрану.

27. React.dev [Електронний ресурс] // React– Режим доступу до ресурсу: <https://react.dev/learn> (дата звернення – 29.04.2023). – Назва з екрану.

28. Djangoproject [Електронний ресурс] // Django– Режим доступу до ресурсу: <https://docs.djangoproject.com/en/4.2/> (дата звернення – 1.05.2023). – Назва з екрану.

29. Ruby on Rails [Електронний ресурс] // Rails – Режим доступу до ресурсу: <https://guides.rubyonrails.org/> / (дата звернення – 4.05.2023). – Назва з екрану.

30. Блог Smashing Magazine [Електронний ресурс] // Smashing – Режим доступу до ресурсу: <https://www.smashingmagazine.com/events/> / (дата звернення – 7.05.2023). – Назва з екрану.

31. Блог CSS-Tricks [Електронний ресурс] // DigitalOcean – Режим доступу до ресурсу: <https://css-tricks.com/guides/> (дата звернення – 10.05.2023). – Назва з екрану.

32. Блог A List Apart [Електронний ресурс] // WordPress – Режим доступу до ресурсу: <https://alistapart.com/> (дата звернення – 13.05.2023). – Назва з екрану.

33. Блог SitePoint [Електронний ресурс] // SitePoint – Режим доступу до ресурсу: <https://www.sitepoint.com/javascript/> (дата звернення – 17.05.2023). – Назва з екрану.

34. Блог Tuts+ [Електронний ресурс] // envatotuts – Режим доступу до ресурсу: <https://tutsplus.com/> / (дата звернення – 20.05.2023). – Назва з екрану.

35. Шарун, А.В. "Програмування на мові С#". Київ: Кондор, 2013. 352 с.

36. Курдюмов, А.Г., Курдюмова, О.А. "Програмування на мові С#. Розробка ігор". Київ: ТОВ "Видавничий дім "Слово", 2017. 416 с.

37. Шварц, О.М., Телепін, А.В. "ASP.NET: Розробка веб-додатків". Київ: ДУХ І ЛІТЕРА, 2012. 460 с.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

38. Жарков, І.М., Мясоєдов, О.М. "ASP.NET MVC: Створення веб-додатків". Київ: Видавництво "Алерта", 2012. 336 с.

39. Стахів, І.М. "Веб-програмування: Підручник". Львів : Світ, 2016. 336 с.

40. Корнійчук, М., "Розробка веб-додатків на платформі ASP.NET". Київ: Центр учбової літератури, 2012. 320 с.

41. Шевченко, О.В. "Програмування на С# для початківців". Київ: Навчальна книга – Богдан, 2019. 400 с.

					<i>КвРІПЗ.190133.19.07.ПЗ</i>	Арк.
						69
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ДОДАТОК А  
(обов'язковий)

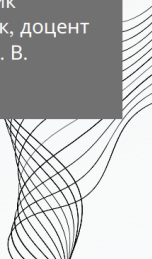
ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ



**СИСТЕМА КЕРУВАННЯ  
КОНТЕНТОМ ВЕБ-  
САЙТУ СОЦІАЛЬНОЇ  
МЕРЕЖІ**

Виконав студент IV курсу,  
група ІПЗ-19-1  
Зайцев Олег

Керівник  
канд. техн. наук, доцент  
Гурман І. В.



Метою проекту є дослідження та аналіз існуючих соціальних мереж та CMS для них, визначення їх переваг та недоліків, розробка власної CMS для соціальної мережі, а також тестування та оцінка її ефективності.

Виходячи із мети, об'єкту та предмету дослідження можна окреслити завдання на дипломний проект:

- дослідити предметну область;
- здійснити аналіз існуючих програмних рішень даної галузі;
- розробити технічне завдання;
- спроектувати розробку програмного забезпечення;
- розробити інструкцію користувача.

# АКТУАЛЬНІСТЬ РОБОТИ

Актуальність роботи полягає у необхідності розробки високоефективних та безпечних CMS для соціальних мереж, що забезпечує стабільну та ефективну роботу веб-сайтів соціальних мереж та покращує їх функціональність. Результати дослідження та розробки нової CMS для соціальної мережі можуть стати корисними для компаній та організацій, які планують розробити свій веб-сайт на основі соціальної мережі. Нова CMS може забезпечити їм ефективну та безпечну роботу сайту, що позитивно позначиться на розвитку їх бізнесу.

## Переваги

Facebook має велику кількість функцій та інструментів, що дозволяють користувачам спілкуватися, ділитися контентом, групуватися за інтересами, відстежувати новини та багато іншого.

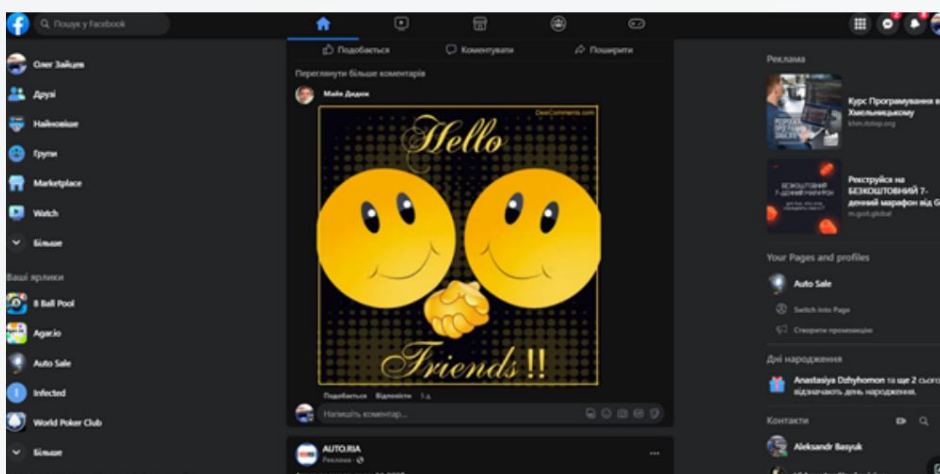
Facebook надає користувачам можливість відстежувати показники взаємодії з аудиторією, такі як кількість лайків, коментарів, репостів та інших взаємодій з контентом.

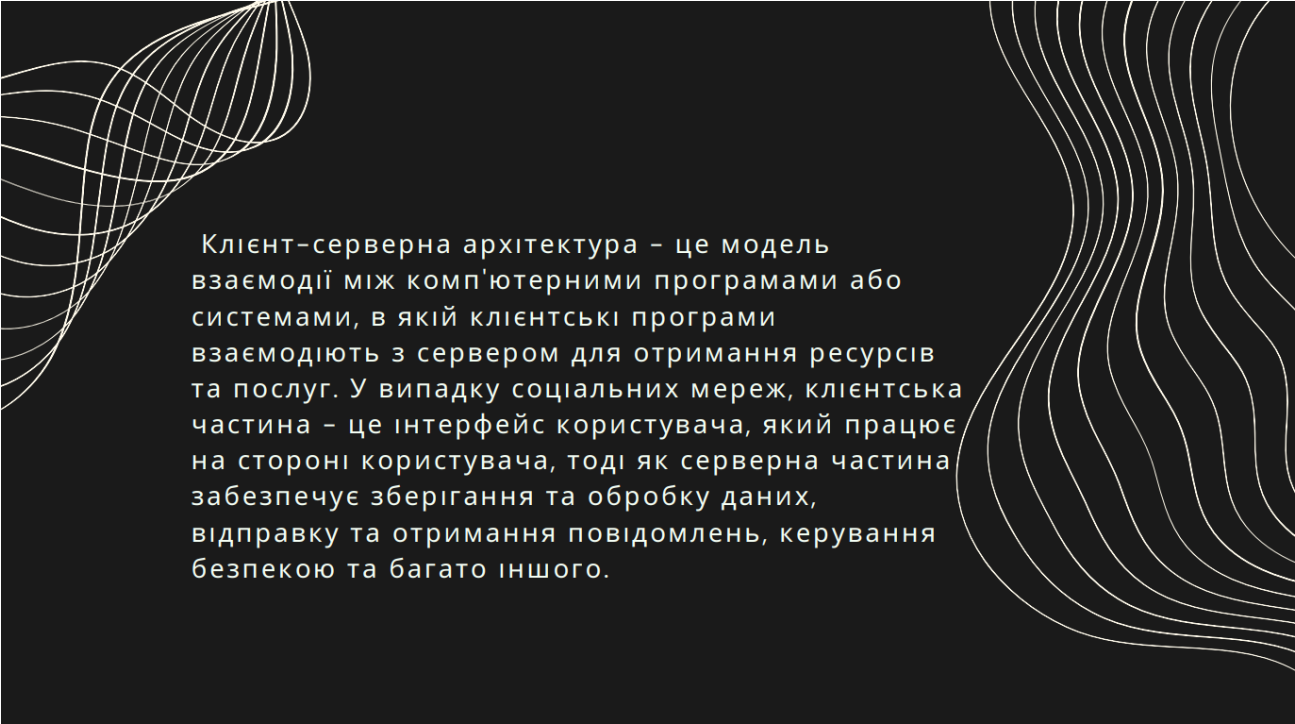
## Недоліки

Реклама та спам. Перенасиченість реклами та спаму на Facebook може стати нав'язливим для користувачів та негативно вплинути на їхній досвід використання платформи.

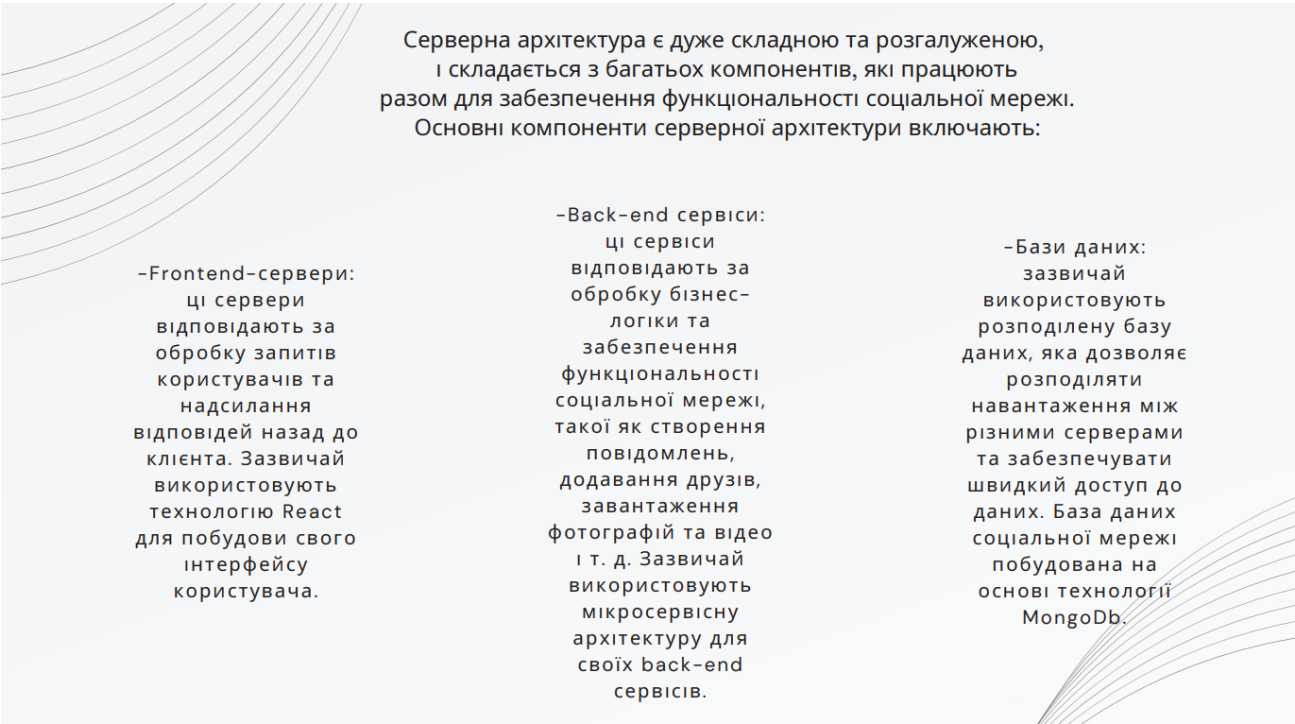
Питання конфіденційності. Facebook був критикований за порушення приватності користувачів та збір їхніх даних без їхньої згоди.

Залежність від алгоритмів. Використання алгоритмів Facebook для відбору та показу контенту може привести до створення "еко-камер" та спотворення поглядів та думок користувачів, що може вплинути на вибори та громадську думку.





Клієнт-серверна архітектура – це модель взаємодії між комп'ютерними програмами або системами, в якій клієнтські програми взаємодіють з сервером для отримання ресурсів та послуг. У випадку соціальних мереж, клієнтська частина – це інтерфейс користувача, який працює на стороні користувача, тоді як серверна частина забезпечує зберігання та обробку даних, відправку та отримання повідомлень, керування безпекою та багато іншого.



Серверна архітектура є дуже складною та розгалуженою, і складається з багатьох компонентів, які працюють разом для забезпечення функціональності соціальної мережі.

Основні компоненти серверної архітектури включають:

-Frontend-сервери:  
ці сервери відповідають за обробку запитів користувачів та надсилання відповідей назад до клієнта. Зазвичай використовують технологію React для побудови свого інтерфейсу користувача.

-Back-end сервіси:  
ці сервіси відповідають за обробку бізнес-логіки та забезпечення функціональності соціальної мережі, такої як створення повідомлень, додавання друзів, завантаження фотографій та відео і т. д. Зазвичай використовують мікросервісну архітектуру для своїх back-end сервісів.

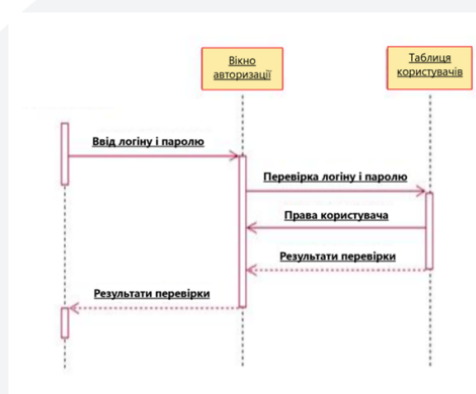
-Бази даних:  
зазвичай використовують розподілену базу даних, яка дозволяє розподіляти навантаження між різними серверами та забезпечувати швидкий доступ до даних. База даних соціальної мережі побудована на основі технології MongoDB.

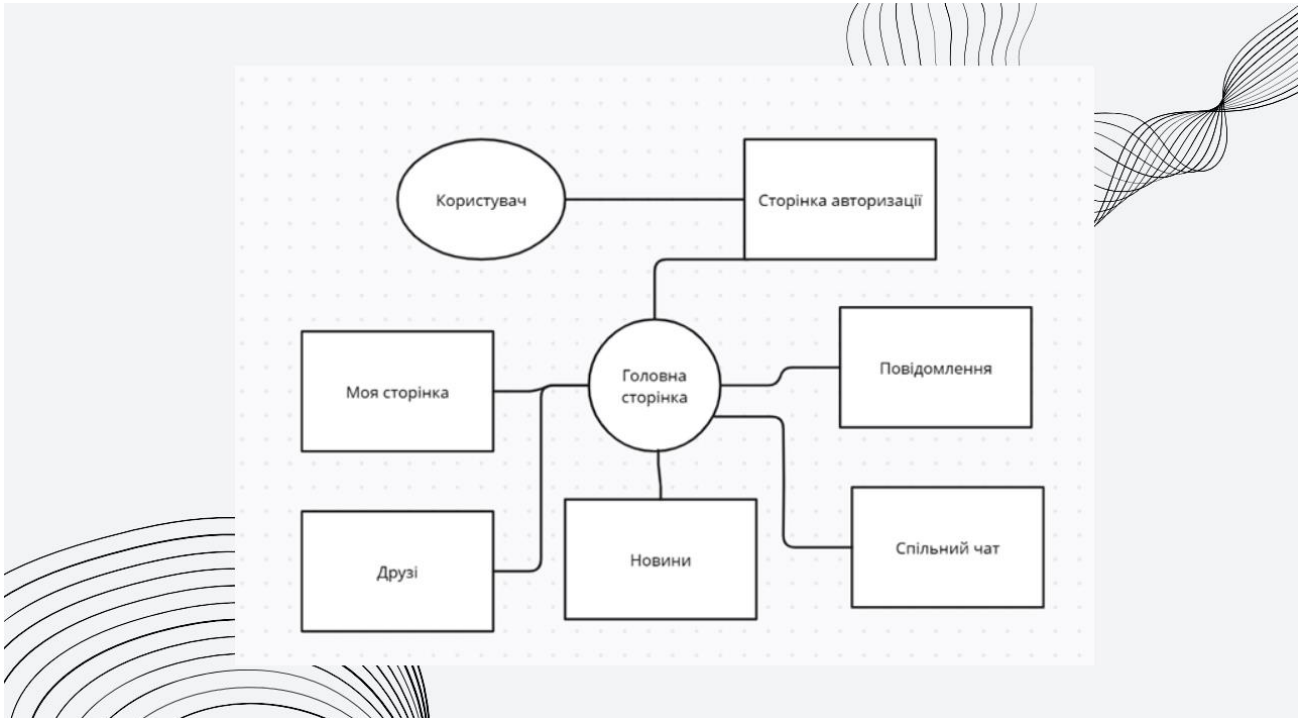
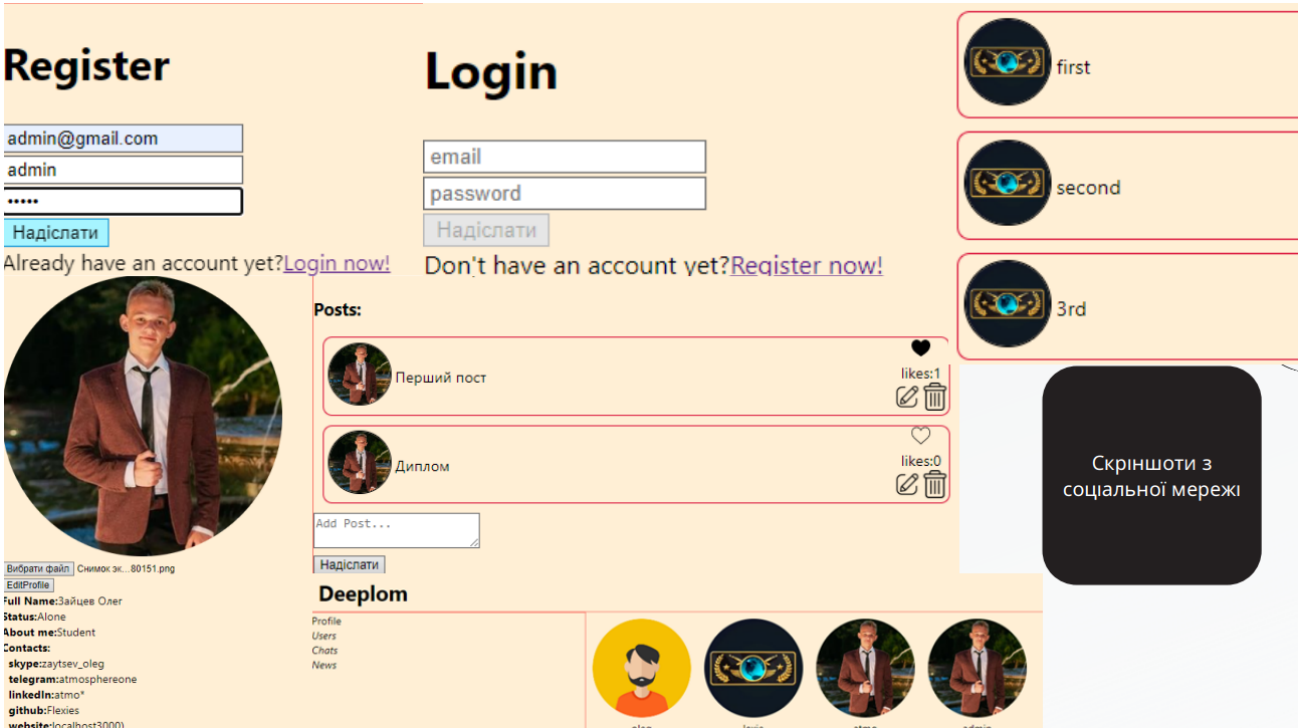
- React має багато переваг порівняно з іншими фреймворками для розробки веб-додатків. Він дозволяє розробникам писати чистий, організований та легко підтримуваний код, покращує продуктивність та ефективність додатку, а також забезпечує швидкий рендерінг сторінок та легке тестування компонентів. Крім того, наявність великої спільноти розробників та багатой екосистеми бібліотек та плагінів дозволяє розширювати функціональність додатку без значних зусиль.

### Вікно авторизації

У функції login оброблюється отриманий запит. Вона отримує з тіла запиту електронну пошту та пароль. Далі, за допомогою методу checkUser з сервісу AuthService, функція перевіряє введені дані з існуючими користувачами в базі даних. Якщо збігу не знайдено, то повертається відповідь з кодом статусу 400 та повідомленням про недійсні облікові дані.

У випадку, якщо користувача знайдено та введені дані вірні, функція встановлює куку з токеном (`res.cookie('token', generateToken(user_id), {maxAge: 60 * 60 * 1000, httpOnly: true})`), що дозволяє зберігати токен аутентифікації на клієнтському боці протягом 60 хвилин.





# ВИСНОВКИ

Під час виконання кваліфікаційної нам вдалося досягти поставленої мети, яка полягала в розробці системи керування контентом для веб-сайту соціальної мережі.

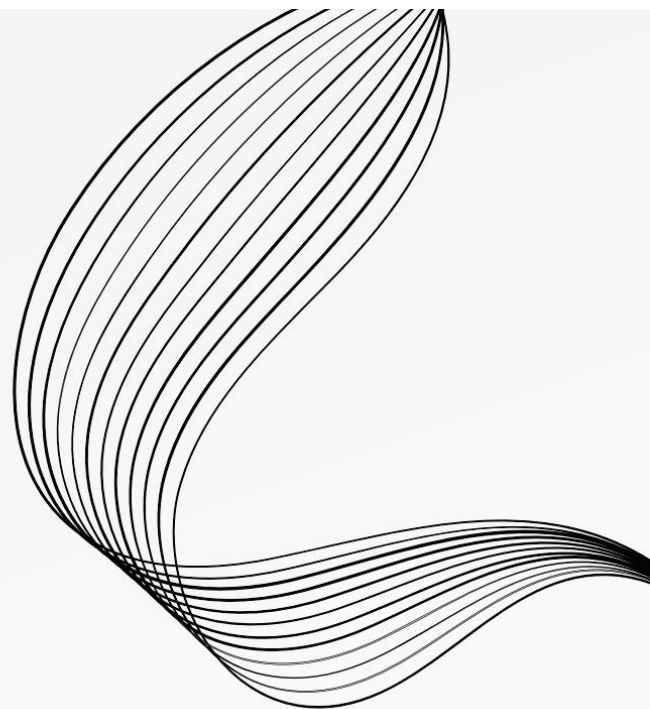
У першому розділі нашої роботи ми провели дослідження предметної області та визначили функціональні та нефункціональні вимоги до системи керування контентом. Також було сформульовано технічне завдання та постановку задачі. Ми також проаналізували наявні рішення у сфері соціальних мереж, виявивши їх позитивні та негативні аспекти.

У другому розділі ми провели проектування соціальної мережі, включаючи архітектуру, структуру, базу даних та інтерфейс користувача. Ми також провели аналіз і вибрали технології та методи реалізації.

У третьому розділі ми здійснили програмну реалізацію соціальної мережі. Ми представили структуру та функціональне призначення модулів програмного засобу, показали реалізацію баз даних та спроектували інтерфейс.

У четвертому розділі ми провели тестування створеного програмного засобу в можливому діапазоні.

# ДЯКУЮ ЗА УВАГУ



## ДОДАТОК Б (обов'язковий)

### ВИГЛЯД ІСНУЮЧИХ РІШЕНЬ

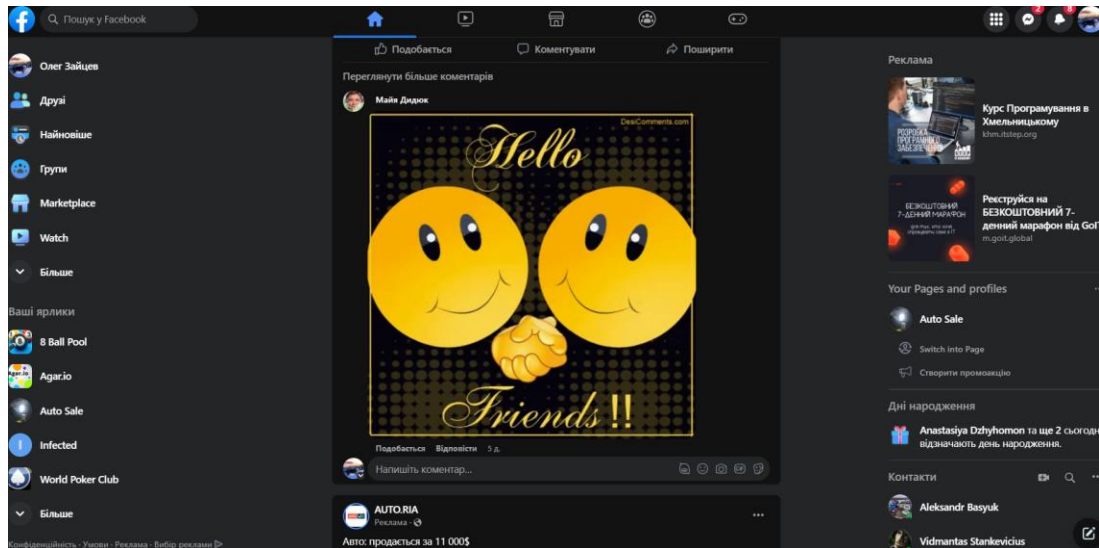


Рисунок В.1 – Вигляд сторінки новин соціальної мережі facebook

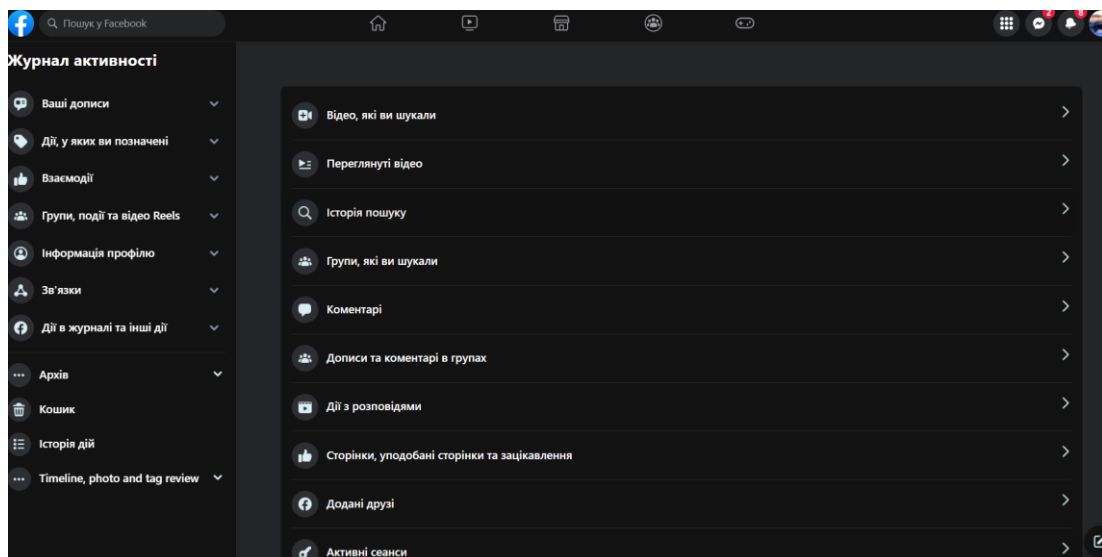


Рисунок В.2 – Вигляд журналу активності соціальної мережі facebook

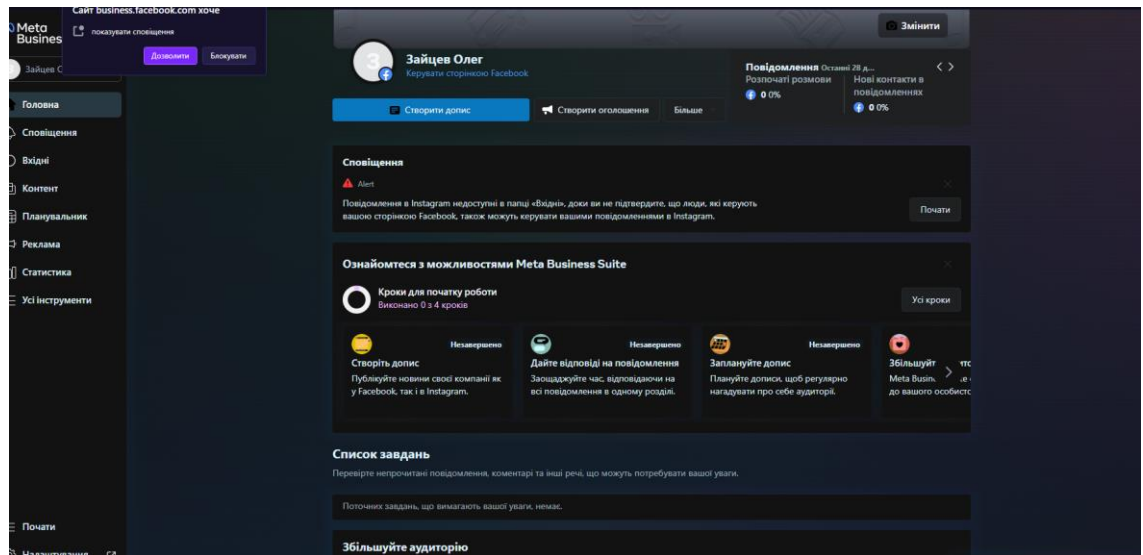


Рисунок В.3 – Вигляд сторінки компанії соціальної мережі facebook



Рисунок В.4 – Вигляд сторінки соціальної мережі facebook у мобільному додатку

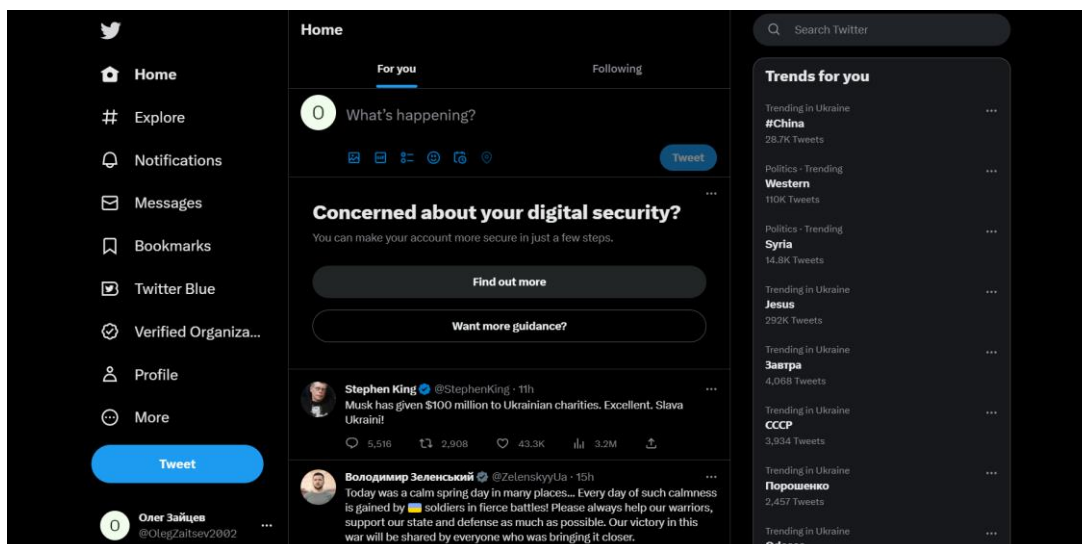


Рисунок В.5 – Вигляд головної сторінки соціальної мережі Twitter

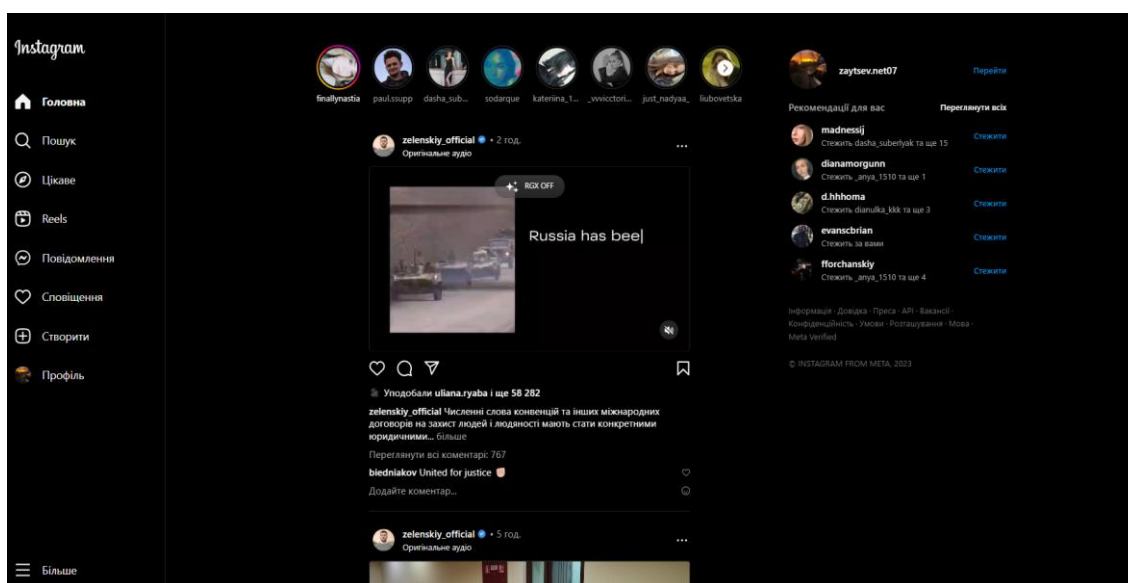


Рисунок В.6 – Вигляд головної сторінки соціальної мережі Instagram

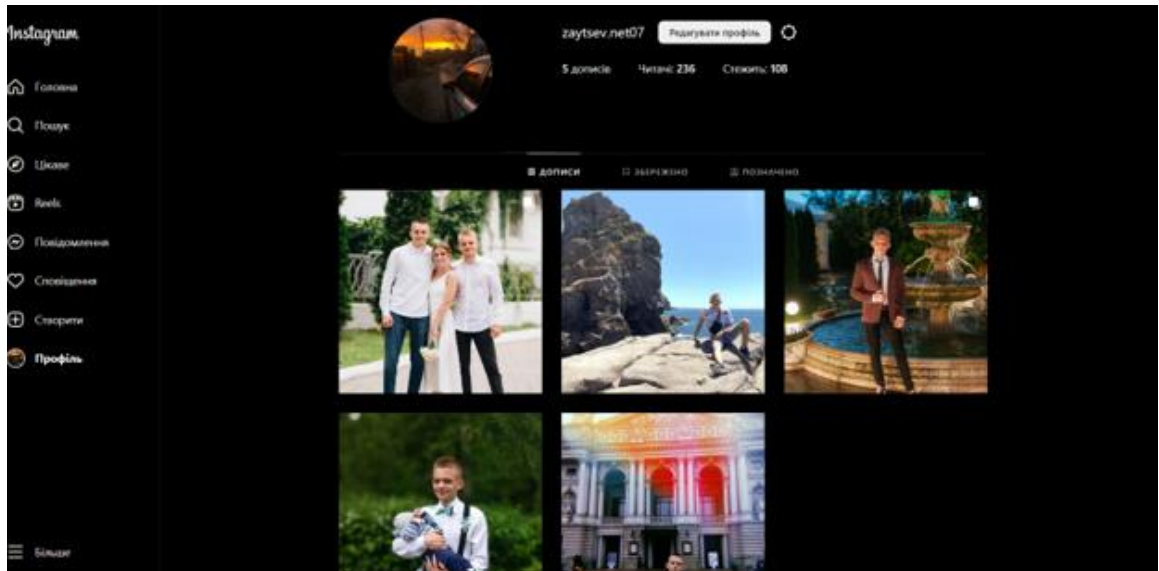


Рисунок 1.7 – Вигляд інформації про головну сторінку соціальної мережі Instagram

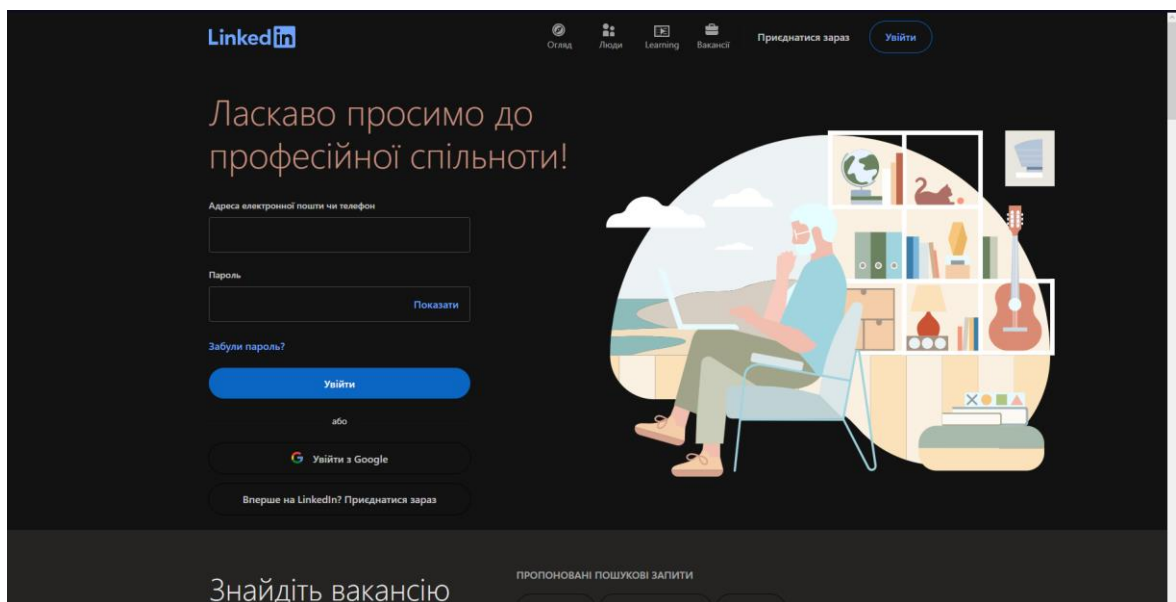


Рисунок 1.8 – Вигляд головної сторінки соціальної мережі LinkedIn

## ДОДАТОК В

### (ОБОВ'ЯЗКОВИЙ)

## ОПИС ОСНОВНИХ МОДУЛІВ СОЦІАЛЬНОЇ МЕРЕЖІ

Модуль проведення автентифікації.

```

const User = require('../models/User')
const jwt = require("jsonwebtoken");
const AuthService = require('../services/AuthService')

const generateToken = (id) => {
  return jwt.sign({id}, process.env.JWT_SECRET, {expiresIn: '1h'})
}

class AuthController {
  //@public
  //POST
  async register(req, res) {
    try {
      const {email, username, password} = req.body
      const candidate = await User.findOne({email})
      if (candidate) return res.status(400).json({message: 'User already
exists'})
      const user = await AuthService.createUser(email, username, password)
      res.cookie('token', generateToken(user._id), {maxAge: 60 * 60 * 1000,
httpOnly: true})
      return res.status(201).json({message: 'Successful registration'})
    } catch (e) {
      return res.status(500).json(e.message)
    }
  }

  //@public
  //POST
  async login(req, res) {
    try {
      const {email, password} = req.body
      const user = await AuthService.checkUser(email, password)
      if (!user) return res.status(400).json({message: 'Invalid
credentials'})
      res.cookie('token', generateToken(user._id), {maxAge: 60 * 60 * 1000,
httpOnly: true})
      return res.status(200).json({message: 'Successful login'})
    } catch (e) {
      return res.status(500).json(e.message)
    }
  }
}

```

```

    }
  }

  //@private
  //DELETE
  async logout(req, res) {
    try {
      if (req.cookies.token) {
        res.clearCookie('token')
        return res.status(200).json({message: 'Successful logout'})
      } else
        return res.status(400).json({message: 'Something gone wrong'})
    } catch (e) {
      return res.status(500).json(e.message)
    }
  }
}

```

Тут описано вихід користувача з наявного облікового запису.

```

//@public
//GET
async me(req, res) {
  try {
    const token = req.cookies.token
    if (!token)
      return res.status(200).json({resultCode: 1, message:
'Unauthorized'})
    const decoded = jwt.verify(token, process.env.JWT_SECRET)
    const user = await User.findById(decoded.id).select(['_id', 'email',
'username'])
    if (!user)
      return res.status(200).json({resultCode: 1, message:
'Unauthorized'})
    return res.status(200).json({resultCode: 0, user})
  } catch (e) {
    return res.status(500).json(e.message)
  }
}
}
module.exports = new AuthController()

```

Описаний процес перевірки авторизації користувача та робота куків.

```

const Post = require('../models/Post')
class PostController {
  //@private
  //POST
  async create(req, res) {
    try {
      const {text} = req.body
      const post = new Post({text, owner: req.user})
    }
  }
}

```

```

        await post.save()
        return res.status(201).json({post})
    } catch (e) {
        return res.status(500).json(e.message)
    }
}
//@private
//PUT
async update(req, res) {
    try {
        const {postId, text} = req.body
        const post = await Post.findById(postId)
        if (post.owner.toString() !== req.user.toString())
            return res.status(400).json({message: 'You can't update this
post'})
        post.text = text
        post.isUpdated = true
        await post.save()
        return res.status(200).json({post})
    } catch (e) {
        return res.status(500).json(e.message)
    }
}
async delete(req, res) {
    try {
        const {id} = req.params//postId
        let post = await Post.findById(id)
        if (post.owner.toString() !== req.user.toString())
            return res.status(400).json({message: 'You can't delete this
post'})
        await post.deleteOne()
        return res.status(200).json({post})
    } catch (e) {
        return res.status(500).json(e.message)
    }
}
}

```

Цей код представляє обробник запиту для видалення допису в соціальній мережі.

```

async like(req, res) {
    try {
        const {postId} = req.body
        let post = await Post.findById(postId)
        if (post.likes.includes(req.user))
            post = await Post.findByIdAndUpdate(postId, {$pull: {likes:
req.user}}, {new: true})
        else
            post = await Post.findByIdAndUpdate(postId, {$push: {likes:
req.user}}, {new: true})
    }
}

```

```

        return res.status(200).json({post})
    } catch (e) {
        return res.status(500).json(e.message)
    }
}

```

Цей код представляє обробник запиту для вподобання допису в соціальній мережі

```

class UserController {
  //@private
  //PUT
  async update(req, res) {
    try {
      const user = await User.findByIdAndUpdate(req.user, {$set: req.body},
        {new: true})
        .select(['-password', '-email'])
        return res.status(200).json({user})
    } catch (e) {
      return res.status(500).json(e.message)
    }
  }
}

```

Цей код представляє обробник запиту для оновлення користувача в соціальній мережі.

```

  async uploadPhoto(req, res) {
    try {
      const {photo} = req.files
      const photoPath = await FileService.saveFile(photo)
      const user = await User.findByIdAndUpdate(req.user, {$set: {photo:
        photoPath}}, {new: true})
        .select(['-password', '-email'])
        return res.status(200).json({user})
    } catch (e) {
      return res.status(500).json(e.message)
    }
  }
}

```

Цей код представляє обробник запиту для завантаження фотографії користувача в соціальній мережі.

```

  async follow(req, res) {
    try {
      const {userId} = req.body
      if (userId === req.user.toString())
        return res.status(400).json({message: 'U can`t follow yourself'})
      let user = await User.findById(userId)
      if (user.followers.includes(req.user)) {
        await User.findByIdAndUpdate(req.user, {$pull: {following:
          userId}}, {new: true})

```

```

        user = await User.findByIdAndUpdate(userId, {$pull: {followers:
req.user}}, {new: true})
      } else {
        await User.findByIdAndUpdate(req.user, {$push: {following:
userId}}, {new: true})
        user = await User.findByIdAndUpdate(userId, {$push: {followers:
req.user}}, {new: true})
      }
      return res.status(200).json({user})
    } catch (e) {
      return res.status(500).json(e.message)
    } } }

```

Цей код відповідає за функціонал "підписки" в соціальній мережі. У функції follow оброблюється отриманий запит. Вона отримує ідентифікатор користувача `userId`, на якого потрібно підписатися, з об'єкта `req.body`.

```

import {useForm} from "react-hook-form";

const RegisterForm = ({registerNewUser}) => {
  const {reset, register, handleSubmit, formState: {errors, isValid}} =
useForm({mode: "onChange"});
  //send data to server and check for errors
  const onSubmit = async (data) => {
    await registerNewUser(data)
    reset()
  }
  return (
    <form onSubmit={handleSubmit(onSubmit)}>
      <div>
        <input type='email' placeholder='email'
          {...register("email", {required: true})} />
        {errors.email && <span>This field is required</span>}
      </div>
      <div>
        <input type='text' placeholder='username'
          {...register("username", {required: true})} />
        {errors.username && <span>This field is required</span>}
      </div>
      <div>
        <input type='password' placeholder='password'
          {...register("password", {required: true})} />
        {errors.password && <span>This field is required</span>}
      </div>
      <input type="submit" disabled={!isValid}/>
    </form>
  );
}
export default RegisterForm

```

Даний фрагмент коду показує візуальну частину, робота front-end для модуля автентифікації. Розміщення блоків та контейнерів, введення даних, перевірка валідності.

```
const Login = () => {
  const isAuthenticated = useSelector(state => state.auth.isAuthenticated)
  const dispatch = useDispatch()
  if (isAuthenticated) return <Navigate to='/profile' />
  return (
    <div>
      <h1>Login</h1>
      <LoginForm login={{data} => dispatch(login(data))} />
      <div>
        Don't have an account yet?
        <NavLink to='/register'>Register now!</NavLink>
      </div>
    </div>
  )
}
```

Цей код відповідає за сторінку входу (логіну) в соціальну мережу.

```
const Profile = () => {
  const profile = useSelector(state => state.user.userProfile)
  const authUserId = useSelector(state => state.auth.id)
  const dispatch = useDispatch()
  let {id} = useParams()
  useEffect(() => {
    if (id) {
      dispatch(getUserProfile(id))
    } else if (authUserId) {
      dispatch(getUserProfile(authUserId))
    }
  }, [id])
  if (!authUserId && !id) return <Navigate to='/login' />
  if (!profile) return <div>loading</div> //TODO: make preloader
  return (
    <div>
      <ProfileInfo isOwner={!id} profile={profile} />
      <MyPosts isOwner={!id} profilePhoto={profile.photo}
authUserId={authUserId} />
    </div>
  )
}
export default Profile
```

Цей код визначає компонент Profile, який відображає інформацію про користувача та його дописи.

## ДОДАТОК Г

### (ОБОВ'ЯЗКОВИЙ)

### ОПИС ПІДКЛЮЧЕННЯ БАЗИ ДАНИХ MongoDB

```
const mongoose = require('mongoose')
const {Schema, model} = mongoose

const postSchema = new Schema({
  text: {type: String, required: true},
  isUpdated: {type: Boolean, default: false},
  likes: [{type: Schema.Types.ObjectId, ref: 'User'}],
  owner: {type: Schema.Types.ObjectId, ref: 'User'}
},
{timestamps: true}
)
module.exports = model('Post', postSchema)
```

Даний код визначає схему для моделі "Post" у MongoDB, використовуючи бібліотеку Mongoose.

```
const mongoose = require('mongoose')
const {Schema, model} = mongoose

const userSchema = new Schema({
  email: {type: String, required: true},
  username: {type: String, required: true},
  password: {type: String, required: true},
  followers: [{type: Schema.Types.ObjectId, ref: 'User'}],
  following: [{type: Schema.Types.ObjectId, ref: 'User'}],
  fullName: {type: String, default: null},
  photo: {type: String, default: null},
  status: {type: String, default: null},
  aboutMe: {type: String, default: null},
  contacts: {
    skype: {type: String, default: null},
    telegram: {type: String, default: null},
    linkedIn: {type: String, default: null},
    github: {type: String, default: null},
    website: {type: String, default: null},
  },
},
{timestamps: true}
)
module.exports = model('User', userSchema)
```

Цей код містить опис структури даних користувача для моделі `mongoose`, що використовується в додатку соціальної мережі.

```
require('dotenv/config')
const express = require('express')
const mongoose = require('mongoose')
const cors = require('cors')
const cookieParser = require('cookie-parser')
const fileUpload = require('express-fileupload')
const authRouter = require('./routes/auth.router')
const userRouter = require('./routes/users.router')
const postRouter = require('./routes/post.router')
const app = express()
app.use(fileUpload({}))
app.use('/static', express.static('static'))
app.use(express.json())
app.use(cookieParser())
app.use(cors({
  origin: process.env.CLIENT_URL,
  credentials: true
}))
app.use('/api/auth', authRouter)
app.use('/api/users', userRouter)
app.use('/api/posts', postRouter)
mongoose.connect(process.env.DB_URI, (error) => {
  if (error) return console.log(error)
  app.listen(process.env.PORT, () => console.log(`Server started on port
${process.env.PORT}`))
  console.log(`Db connected successfully`)
})
```

Цей код є основним файлом додатку, який налаштовує підключення до бази даних MongoDB.

## ДОДАТОК Д (обов'язковий)

### ТЕХНІЧНЕ ЗАВДАННЯ

#### **Введення**

Робота виконується в рамках проекту розробки веб-застосунку для продажу будівельних інструментів, у вигляді сайту, що дозволяє здійснювати онлайн покупки.

#### **1 Підстава для розробки**

Підставою для розробки даного веб-застосунку є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри програмного забезпечення. Найменування розробки: «Веб-застосунок для продажу комп'ютерної периферії».

#### **2 Призначення розробки**

##### **2.1 Функціональне призначення**

Функціональне призначення веб-застосунку - надати клієнтам можливість перегляду та придбання комп'ютерної периферії через персональний комп'ютер чи смартфон. Додаток також забезпечує можливість оплати товарів в електронному вигляді та виступає посередником між клієнтами та працівниками магазину.

##### **1.2. Експлуатаційні призначення**

Веб-застосунок має використовуватись звичайними користувачами без професійної підготовки за допомогою браузера на ПК, ноутбуках, планшетах та смартфонах, підтримуючи операційні системи Windows, Mac, Android та iOS.

### **3 Вимоги до програми**

#### **3.1 Вимоги до функціональних характеристик**

Веб-застосунок для продажу будівельних інструментів має забезпечувати наступні функції:

- додавання товарів до каталогу;
- реєстрація та авторизація у системі;
- налаштування особистого профілю(редагування даних, зміна пароллю);
- відображення товарів та інформації про них;
- фільтрація для пошуку товарів;
- додавання товарів до кошику;
- оформлення замовлення.

#### **3.2 Вимоги до надійності**

Веб-застосунок повинен забезпечувати захист даних користувачів та передавати особисті дані клієнтів і інформацію про їх спосіб оплати тільки в зашифрованому вигляді. Система повинна відображати повідомлення про помилку та перенаправляти користувача на належно функціонуючу сторінку при виникненні помилок.

#### **3.3 Умови експлуатації**

Умови експлуатації мають відповідати санітарним і технічним нормам експлуатації персонального комп'ютера, при температурі та відносній вологості навколишнього середовища, визначених для персональної обчислювальної техніки згідно з ГОСТ 15150-69 [7].

Для обслуговування веб-застосунку допускаються тільки спеціально навчені адміністратори або розробники. До користування веб-застосунком допускаються користувачі мережі Інтернет.

### **3.4 Вимоги до інформаційної та програмної сумісності**

Для створення клієнтської частини веб-застосунку використовувалася мова розмітки HTML, каскадні таблиці стилів CSS та мова програмування JS

Для створення серверної частини веб-застосунку використовувалися технології платформи ASP.NET, мова програмування C# та фреймворки ASP.NET MVC і Entity Framework

### **3.5 Спеціальні вимоги**

Для коректної функціональності ПЗ спеціальні вимоги відсутні.

## **4 Вимоги до програмної документації**

Набір документації, що надається у момент здачі проекту:

Опис модулів програми з демонстрацією їх роботи

Текст програми з коментарями

Керівництво користувача для звичайних користувачів та адміністратора

Довідкова інформація

Керівництво для розробника

## **5 Стадії та етапи розробки**

Стадії та етапи розробки веб-застосунку для продажу будівельних інструментів продемонстровано у таблиці 1

Таблиця 1 - стадії та етапи розробки

Стадія розробки	Етапи робіт	Зміст робіт
Технічне завдання 02.01 – 31.01.2023	Обґрунтування необхідності розробки програми	Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання
Ескізний проект 01.02 – 14.02 2023	Розробка ескізного проекту	Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури системи, що буде розроблюватися
Технічний проект 15.02 – 28.02 2023	Розробка технічного проекту	Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми; остаточне визначення конфігурації технічних засобів
Робочий проект 01.03 – 10.04.2023	Розробка програмного забезпечення	Реалізація програмного забезпечення; відладка; проведення попереднього тестування
Розробка програмної документації 11.04 – 20.04.2023	Розробка документації до програмного забезпечення	Розробка необхідної документації, передбаченої технічним завданням
Тестування системи 21.04 – 30.04.2023	Проведення тестування програмного забезпечення	Розробка методики тестування; проведення основних тестів; коректування програмного забезпечення
Впровадження	Підготовка і передача програми	Підготовка і передача програмного забезпечення; навчання персоналу використуванню програмного забезпечення; внесення коректувань в програмне забезпечення і документацію

## 6 Порядок контролю та приймання

Контроль здійснюється користувачами застосунку підключеними на етапі тестування. Прийом програми здійснюється після повного її розгортання, тестування і налаштування для справного функціонування

ДОДАТОК Е  
(обов'язковий)

ПРОХОДЖЕННЯ АНТИПЛАГІАТУ

**Anti-Plagiarism v-15.257**

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 10%

ID: 114564 Назва: БКР Система керування контентом веб-сайту соціальної мережі Додано в БД: 2023-06-02 Автора: Зайцев О.О. Керівники: Гурман І.В. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	98263	817	2850 (3%)	37 (5%)



Ім'я користувача:  
Кафедра ІПЗ

Дата перевірки:  
02.06.2023 12:36:59 EEST

Дата звіту:  
02.06.2023 12:39:19 EEST

ID перевірки:  
1015390652

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100005589

Назва документа: БКР Зайцев ІПЗ-19 плагіат

Кількість сторінок: 71 Кількість слів: 14419 Кількість символів: 109852 Розмір файлу: 1.18 MB ID файлу: 1015055195

**5.2%**  
**Схожість**

Найбільша схожість: 2.12% з джерелом з Бібліотеки (ID файлу: 1015054441)

3.26% Джерела з Інтернету 423 ..... Сторінка 73

3.58% Джерела з Бібліотеки 84 ..... Сторінка 75

**0% Цитат**

Вилучення цитат вимкнено

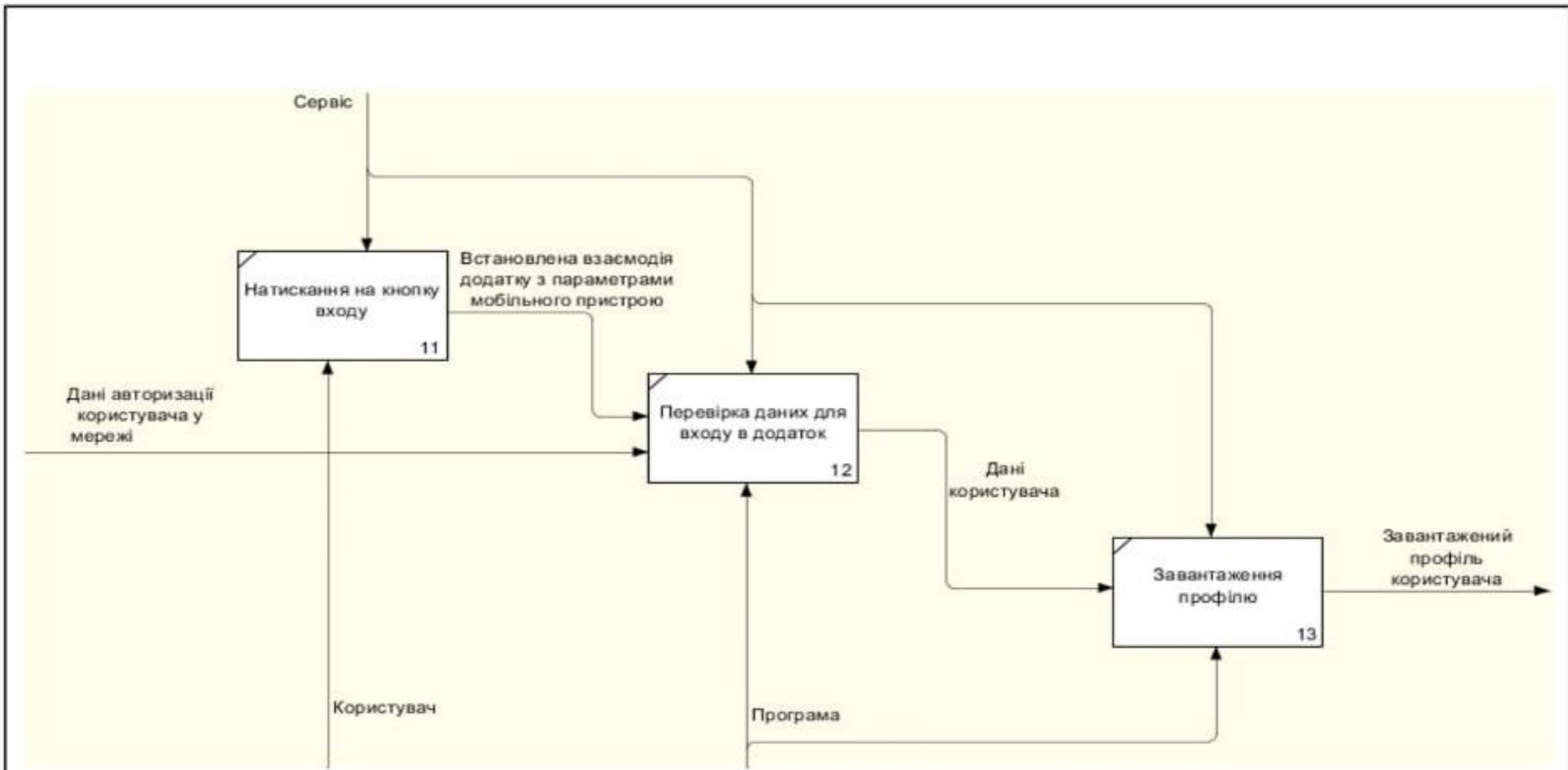
Вилучення списку бібліографічних посилань вимкнено

**0%**  
**Вилучень**

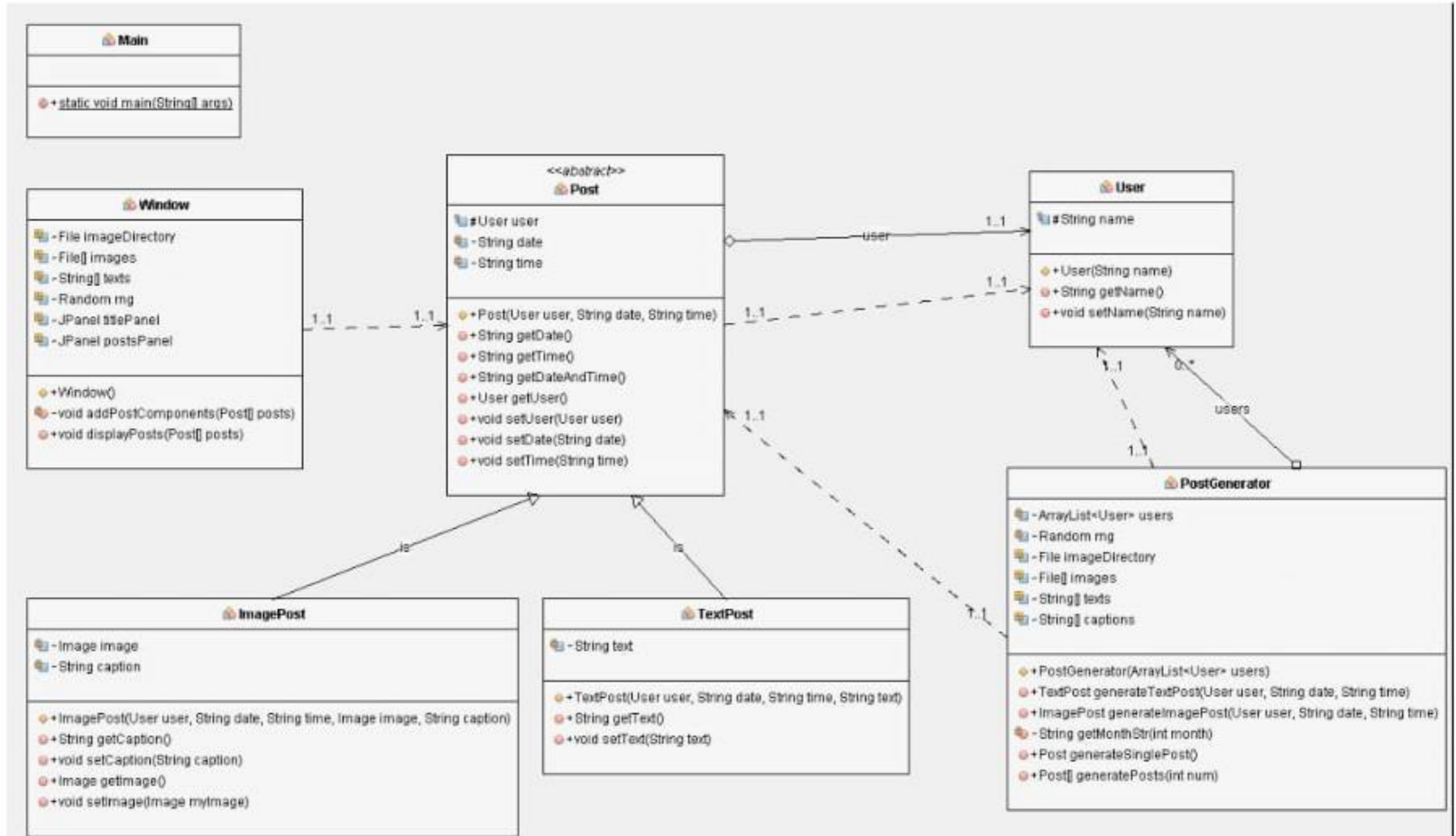
Немає вилучених джерел

ДОДАТОК Є  
(Обов'язковий)

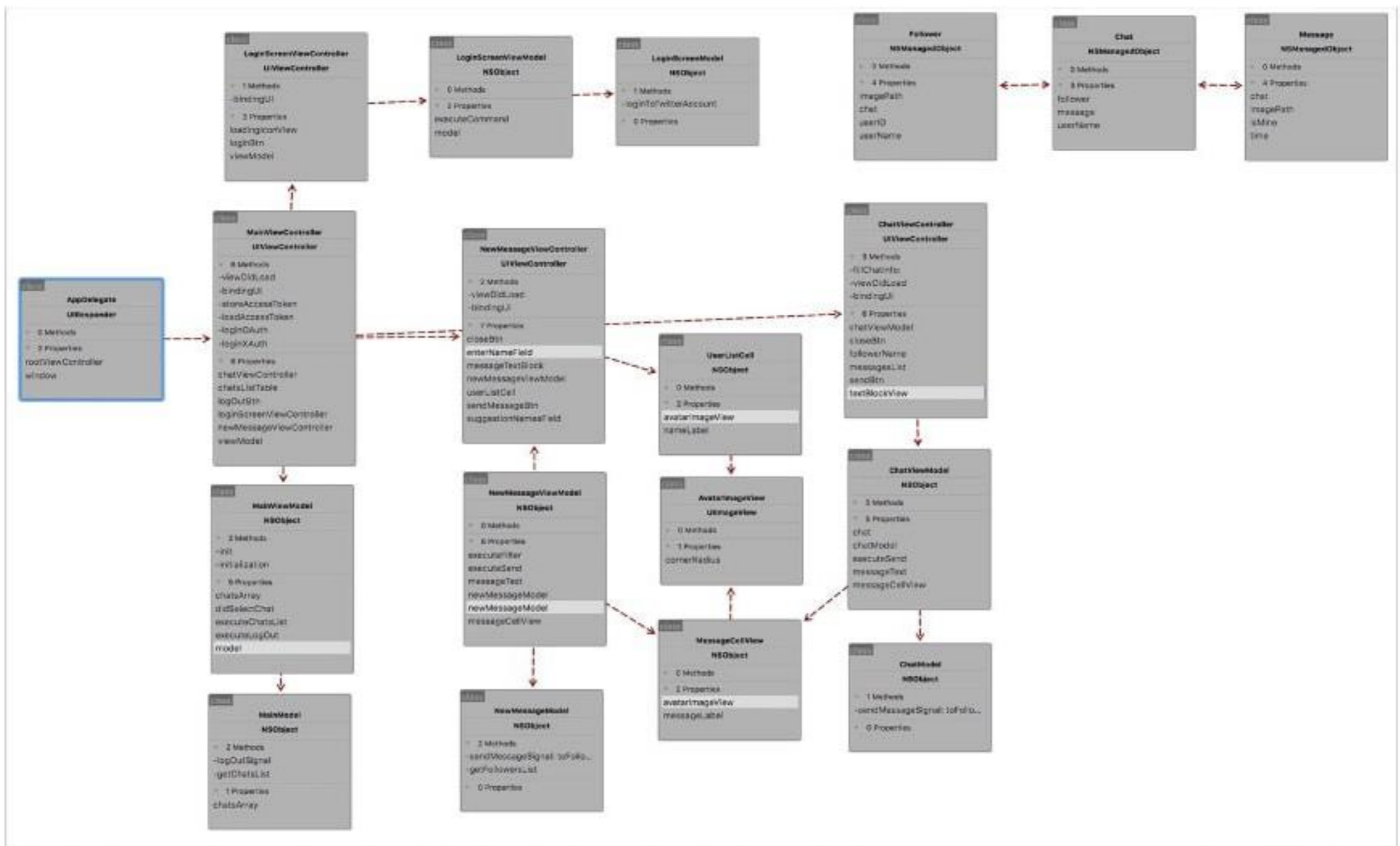
**ГРАФІЧНІ МАТЕРІАЛИ**



					<b>КвРІПЗ_190133.01.07.E8</b>		
					UML-діаграма декомпозиції першого рівня		
					Літера	Маса	Масштаб
<u>Зм.</u>	<u>Арк.</u>	<u>№ докум.</u>	<u>Підпис</u>	<u>Дата</u>			
Розробив		Зайцев О. О.					
Керівник		Гурман І. В.					
<u>Консульт.</u>					Аркуш 1	Аркушів 3	
<u>Н. Контр.</u>		Федун Ю. В.					
<u>Зав. Квф.</u>		Белратюк Л.П.					



					<b>КвРІПЗ 190133.01.07.Е8</b>		
					<b>ER-діаграма бази даних</b>		
					Літера	Маса	Масштаб
					Аркуш 2	Аркушів 3	
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Зайцева О. О.					
Керівник		Турман І. В.					
Консульт.							
Іл. Контр.		Розум І. В.					
Зав. Кад.		Бедрагір Л. П.					



				<b>КвРІПЗ 190133.01.07.Е8</b>		
				<b>ER-діаграма класів</b>		
Зм. Арх.	№ докум.	Підпис	Дата	Літера	Маса	Масштаб
Розробив	Зайцев О. О.					
Керівник	Гурман І. В.					
Консульт.						
Н. Конст.	Фракун Ю. В.					
Зав. Каф.	Бендратік Л. П.					
				Аркуш 3		Аркушів 3



# Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 1.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. **Помилки в документах: 10%**

ID: 114564 Назва: БКР Система керування контентом веб-сайту соціальної мережі Додано в БД: 2023-06-02 Автора: Зайцев О.О. Керівники: Гурман І.В. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	98263	817	2850 (3%)	37 (5%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:  
Кафедра ІПЗ

ID перевірки:  
1015390652

Дата перевірки:  
02.06.2023 12:36:59 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
02.06.2023 12:39:19 EEST

ID користувача:  
100005589

Назва документа: БКР Зайцев ІПЗ-19 плагіат

Кількість сторінок: 71 Кількість слів: 14419 Кількість символів: 109852 Розмір файлу: 1.18 MB ID файлу: 1015055195

## 5.2% Схожість

Найбільша схожість: 2.12% з джерелом з Бібліотеки (ID файлу: 1015054441)

3.26% Джерела з Інтернету 423 ..... Сторінка 73

3.58% Джерела з Бібліотеки 84 ..... Сторінка 75

## 0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

## 0% Вилучень

Немає вилучених джерел

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ  
освітнього ступеня «Бакалавр»

Дипломник Зайцев Олег Олександрович

Тема Система керування контентом веб-сайту соціальної мережі

Спеціальність 121 – Інженерія програмного забезпечення

**Обсяг кваліфікаційної роботи:**

Кількість листів креслень 3; кількість сторінок записки 68

1. Короткий зміст пояснювальної записки та прийнятих рішень. У дипломній роботі досліджено і проаналізовано предметну область, визначено функціональні та нефункціональні вимоги, проведено аналіз відомих на ринку рішень і доведено актуальність розроблення нового програмного забезпечення. Розглянуто інструменти, які використані для реалізації рішення, та розроблено нове програмне забезпечення для керування контентом веб-сайту соціальної мережі. Проведено тестування розробленого програмного забезпечення, за результатами якого доведено, що програмне забезпечення функціонує коректно та готове до експлуатації.

2. Висновок про відповідність роботи поставленому завданню. Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи. У вступі доведено актуальність теми, визначено мету та завдання дипломного проєктування. У першому розділі проведено аналіз предметної області та розглянуто існуючі програмні рішення; визначено функціональні і нефункціональні вимоги до розроблюваного програмного забезпечення. У другому розділі проведено аналіз переваг і недоліків сучасних архітектур та визначено архітектуру, яка буде використовуватися при розробці нового програмного забезпечення. У третьому розділі представлено розроблені програмні модулі системи керування контентом веб-сайту соціальної мережі та наведено опис їх особливостей. Також проведено тестування програмного забезпечення.

4. Позитивні сторони роботи. Тематика кваліфікаційної роботи є актуальною, оскільки сьогодні системи керування контентом веб-сайтів соціальних мереж є досить популярними. Під час розроблення застосунку було враховано недоліки відомих рішень, а також застосовано нові технології, такі як react та node.js.

5. Негативні сторони роботи У роботі реалізовано не достатньо повний список налаштувань веб-сайту соціальної мережі, таким чином, соціальна мережа розрахована на обмежене коло користувачів.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконане відповідно до теми кваліфікаційної роботи та подане у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки; матеріал пояснювальної записки структурований та послідовний, що дозволяє зрозуміти викладений матеріал у межах тематики проєктування. Графічний матеріал надає можливість наочно побачити деталі проєктування системи керування контентом веб-сайту соціальної мережі.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ Нічепорук Андрій Олександрович, кандидат технічних наук, доцент кафедри комп'ютерної інженерії та інформаційних систем (КІІС) ХНУ

“ 5 ” червог 2023 р.

  
(підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів подібності щодо роботи, продукованими програмно-технічним засобом (ами) перевірки текстів на плагіат:

Назва: «Система керування контентом веб-сайту соціальної мережі»

Автор: Зайцев Олег Олександрович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Гурман Іван Васильович, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданій поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титульний аркуш, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

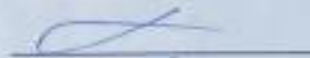
3) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/ схожості, складає 5,2% і адресується до 423 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

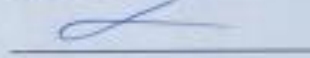
Дата 3.06.2023р

Завідувач кафедри



Леонід БЕДРАТЮК

Гарант освітньої програми



Леонід БЕДРАТЮК

Керівник кваліфікаційної роботи



Іван ГУРМАН