

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

на тему: «Розробка веб-додатку соціальної мережі»

Смірнов Владислав Романович

(прізвище, ім'я та по батькові магістра повністю)

Київ 2023 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ

Тетяна Гончаренко
„__” _____ 2023 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

на тему: «Розробка веб-додатку соціальної мережі»

Виконав: студент 2 – го курсу, групи КНм – 22

Спеціальності: 122 «Ком'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Магістрант Смірнов Владислав Романович .
(прізвище та ініціали)

Керівник Поплавський Олександр Анатолійович .
(прізвище та ініціали)

Рецензент Баліна Олена Іванівна .
(прізвище та ініціали)

Київ, 2023 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій .
Кафедра: інформаційних технологій .
Освітній рівень: «магістр за ОПП» .
Спеціальність: 122 «Комп'ютерні науки» .

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ

Тетяна Гончаренко
„___” _____ 2023 року

З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

-
1. Тема роботи: Розробка веб-додатку соціальної мережі.
затверджена наказом ректора КНУБА № 1280/2 від «26» Червня 2023р.
 2. Керівник роботи: Поплавський Олександр Анатолійович.
 3. Строк подання студентом роботи до захисту: 9 Грудня 2023р.
 4. Зміст пояснювальної записки за розділами:
 - Р.1. Аналіз предметної області.
 - Р.2. Проектування програмного забезпечення.
 - Р.3. Розробка програмного забезпечення веб-додатку.
 - Р.4. Тестування інтерфейсу програмного забезпечення.
 5. Інформаційні слайди:
 - С.1. Вступ.
 - С.2. Аналіз предметної області.
 - С.3. Постановка задачі.
 - С.4. – С.5. Аналіз існуючих рішень.

- С.6. Діаграма варіантів використання.
- С.7. Діаграма діяльності.
- С.8. Діаграма послідовності.
- С.9. Концептуальна модель БД.
- С.10. Вибір засобів розробки.
- С.11. – С. 15. Результати розробки.
- С.16. Висновок

6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р.1. Аналіз предметної області.	Вересень 2023
Р.2. <u>Проектування програмного забезпечення.</u>	Жовтень 2023
Р.3. <u>Розробка програмного забезпечення веб-додатку.</u>	Грудень 2023
Р.4. <u>Тестування інтерфейсу програмного забезпечення.</u>	Грудень 2023
Остаточне оформлення роботи	Грудень 2023
Направлення роботи на рецензування, перевірку на плагіат	Грудень 2023
Попередній захист роботи на кафедрі	Грудень 2023

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірив	
		дата	підпис
Розділ 1.			
Розділ 2.			
Розділ 3.			
Розділ 4.			
Розділ 5.			

8. Дата видачі завдання: «26» Червня 2023р.

Керівник

(підпис)

Поплавський О.А.

(прізвище та ініціали)

Магістрант

(підпис)

Смірнов В.Р.

(прізвище та ініціали)

РЕЗЮМЕ

Київський національний університет будівництва і архітектури

Смірнов Владислав Романович

факультет автоматизації і інформаційних технологій,

група КНм-22

Тема атестаційної випускної роботи: «Розробка веб-додатку соціальної мережі»

освітній рівень: магістр,

спеціальність: 122 «Комп'ютерні науки»,

Науковий керівник: Поплавський Олександр Анатолійович

Обсяг роботи. Атестаційна випускова робота магістра складається: розділів 4, стор. 114, таблиць 2, рис. 56, завдання, анотація, вступу, висновків, списку використаних джерел та додатків.

Актуальність теми. Оскільки соціальні мережі відіграють важливу роль в житті кожної людини, розробка власної соціальної мережі стає важливою завданням, вона дасть можливість користувачам отримати персоналізований та безпечний простір для спілкування, обміну інформацією та творчої активності. Заснована на новаторських технологіях, магістерська робота має на меті розробити веб-додаток соціальної мережі, який враховує сучасні тенденції та вимоги користувачів.

У вступі визначені основні завдання дослідження. Визначені завдання включають в себе дослідження технологій, розробку архітектури системи та тестування розробленого додатку.

У першому розділі «Аналіз предметної області» відбувається вивчення предметної області, де визначено ключові аспекти соціальних мереж, їхню роль у сучасному інформаційному просторі та різноманітність переваг та недоліків, пов'язаних із їх використанням. З урахуванням цього аналізу була визначена мета та поставлені завдання для подальшої розробки соціальної мережі.

У другому розділі «Проектування програмного забезпечення» відбувається глибокий аналіз, та проектування веб-додатку соціальної мережі, використовуючи структуровану методологію та UML-діаграми. Розділ визначає чіткі цілі та завдання, будує ефективну концептуальну модель, а також враховує важливі аспекти системи, такі як взаємодії користувачів та структура об'єктів.

У третьому розділі «Розробка програмного забезпечення веб-додатку» розкривається важливий вибір технічних рішень, включаючи середовище розробки, мову програмування, платформу, бібліотеки та базу даних. Кожен з цих виборів був обґрунтований врахуванням вимог щодо ефективності, зручності розробки та надійності.

У четвертому розділі «Тестування інтерфейсу програмного забезпечення» проведено тестування готового веб-додатку, де визначено його високий рівень функціональності та якості. Результати тестування дозволили виявити та усунути всі помилки, гарантуючи стабільну та безперебійну роботу соціальної мережі.

Ключові слова: соціальна мережа, веб-додаток, фреймворк, база даних, інтерфейс, допис.

Якість оформлення проекту. Атестаційна випускна робота магістра оформлена у відповідності до діючих нормативних документів та методичних вказівок до виконання дипломних робіт для студентів спеціальності 122 «Комп'ютерні науки». Порухень та зауважень під час розробки та перевірки дипломної роботи не виявлено.

Загальний висновок стосовно роботи та присвоєння авторіві освітньо-кваліфікаційного рівня «магістр». Робота виконана якісно та на високому рівні, студент продемонстрував достатній рівень теоретичної підготовки та сформованих практичних навичок в області сучасних інформаційних технологій. Заслуговує оцінки «Відмінно».

Науковий керівник _____ / Поплавський Олександр Анатолійович.
(підпис)

Посада, місце роботи: доцент кафедри інформаційних технологій КНУБА
«4» Грудня 2023р.

АНОТАЦІЯ

Смірнов В.Р. «Розробка веб-додатку соціальної мережі».

Атестаційна випускна робота магістра за спеціальністю: 122 «Комп'ютерні науки». – Київський національний університет будівництва та архітектури. – Київ, 2023 р.

Атестаційна робота магістра присвячена розробці веб-додатку соціальної мережі. Результатом розробки є соціальна мережа, яка вирізняється дбайливою увагою до приватності користувачів та захисту їх особистих даних, вона надає можливість користувачам можливість активної взаємодії та спілкування один з одним, обміну інформацією.

Соціальна мережа наведена в цій роботі є захищеною від зовнішнього втручання та привабливою для користувачів.

Ключові слова: соціальна мережа, веб-додаток, фреймворк, база даних, інтерфейс, допис.

ANNOTATION

Smirnov V.R. "Development of a Social Network Web Application".

Master's Degree Certification Thesis in the specialty: 122 "Computer Sciences". – Kyiv National University of Construction and Architecture. – Kyiv, 2023.

This master's thesis is dedicated to the development of a web-based social network application. The result of the development is a social network that stands out for its careful attention to user privacy and the protection of their personal data. It provides users with the opportunity for active interaction and communication with each other, as well as data exchange.

The social network described in this work is secure against external interference and attractive to users.

Keywords: social network, web application, framework, database, interface, post.

РЕЦЕНЗІЯ

на атестаційну випускную роботу

Студента Смірнова Владислава Романовича

Факультет автоматизації і інформаційних технологій

спеціальності 122 «Комп'ютерні науки»

Тема роботи: Розробка веб-додатку соціальної мережі.

Обсяг роботи: атестаційна випускова робота магістра складається: розділів 4, стор. 114, таблиць 2, рис. 56, завдання, анотація, вступу, висновків, списку використаних джерел та додатків.

Висновок про відповідність завданню: робота виконана у повній відповідності до завдання і у встановлений термін .

Актуальність обраної теми: обрана тема дослідження є актуальною оскільки соціальні мережі є невід'ємною частиною життя багатьох людей. За допомогою соціальних мереж, користувачі можуть зв'язуватися з іншими людьми, обмінюватися інформацією та розважатися. Розробка веб-додатку соціальної мережі може забезпечити нові можливості для користувачів в спілкуванні та обміні інформацією.

Використання у роботі сучасних досягнень науки і техніки: розробка проекту базується на використанні сучасних інформаційних комп'ютерних технологій

Використання у роботі комп'ютерних технологій: Visual Studio Code, Mermaid, Draw.io.

Практичне значення роботи: впровадження сучасних інформаційних технологій має забезпечувати виконання ряду вимог, у тому числі наявність зручного і дружнього інтерфейсу, забезпечення безпеки за допомогою різних методів контролю та розмежування доступу до інформаційних ресурсів, підтримку розподіленої обробки інформації.

Якість оформлення роботи: випускна робота оформлений у відповідності до діючих нормативних документів та методичних вказівок для студентів спеціальності 122 «КН»

Зауваження та побажання: Зауважень не виявлено

Загальний висновок стосовно роботи та надання авторові освітнього ступеня "магістр": робота виконана на високому рівні, студент продемонстрував високий рівень теоретичної підготовки та сформованих практичних навичок в області сучасних інформаційних технологій. Заслуговує оцінки «відмінно».

Рецензент _____ / _____ доц., к.т.н., Баліна О.І. /
(підпис) (науковий ступінь, вчене звання,
прізвище та ініціали)

Посада, місце роботи: доцент кафедри інформаційних технологій проєктування та прикладної математики КНУБА

«4» Грудня 2023р.

ЗМІСТ

ВСТУП	12
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	13
1.1 Виявлення загальних переваг та недоліків соціальних мереж	13
1.2 Аналіз існуючих рішень	16
1.2.1 Instagram	16
1.2.2 Facebook	20
1.2.3 Twitter(x)	23
1.3 Порівняння існуючих рішень з розроблюваною системою	26
1.4 Постановка задачі	27
<i>Висновок ДО РОЗДІЛУ 1</i>	<i>28</i>
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	29
2.1 Визначення цілей, дерево цілей.	29
2.2 Розробка технічного завдання	31
2.2.1 Data Flow Diagram	32
2.2.2 Концептуальна модель	35
2.3 UML діаграми прецедентів, класів, послідовності, діяльності	38
2.3.1 Діаграма прецедентів	38
2.3.2 Діаграма класів	41
2.3.3 Діаграма послідовності	47
2.3.4 Діаграма діяльності	50
<i>Висновок ДО РОЗДІЛУ 2</i>	<i>52</i>
3 Розробка програмного забезпечення ВЕБ-додатку соціальної мережі	53
3.1 Вибір засобів розробки	53
3.1.1 Огляд середовища розробки	53
3.1.2 Огляд обраних мов програмування та методологій	54
3.1.3 Огляд програмної платформи	59
3.1.4 Огляд використаних бібліотек та фреймворків	61
3.1.5 Огляд обраної бази даних	71
3.2 Опис модулів серверної частини веб-додатку соціальної мережі	73
3.2.1 Server.js	73
3.2.2 authRouter.js	74
3.2.3 authCtrl.js	75
3.2.4 auth.js	78
3.2.5 userModel.js	80
<i>Висновок ДО РОЗДІЛУ 3</i>	<i>83</i>

4	ТЕСТУВАННЯ ІНТЕРФЕЙСУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	84
	Висновок ДО РОЗДІЛУ 4	101
	Висновок	102
	Список використаних джерел	103
	ДОДАТОК А – ІНФОРМАЦІЙНІ СЛАЙДИ	106

ВСТУП

В сучасному світі, інформаційні технології та мережа Інтернет перетинають усі сфери нашого життя, змінюючи їх завжди й назавжди. Один з найбільш суттєвих внесків в цю цифрову революцію полягає в розробці веб-додатків, зокрема соціальних мереж. Сучасні соціальні мережі стали не лише платформою для зв'язку та обміну інформацією, але й важливим фактором формування культури, взаємозв'язків і впливу на суспільство.

Ця АВР присвячена розробці веб-додатку соціальної мережі, і вона ставить перед собою завдання дослідити, розробити та реалізувати платформу, яка відповідає сучасним потребам користувачів в онлайн-спільноті. Під час розробки цього веб-додатку буде приділена особлива увага забезпеченню приватності користувачів та захисту їх особистих даних, а також розробці функціоналу, який сприятиме активній взаємодії та обміну інформацією.

Робота над проектом дозволить не лише вивчити та впровадити сучасні веб-технології та мови програмування, але й дослідити важливий аспект розробки соціальних мереж – забезпечення безпеки користувачів.

Результатом цієї роботи буде функціональний веб-додаток, який може стати важливим внеском у сферу онлайн-спільнот і відкрити нові можливості для спілкування та взаємодії в цифровому світі.

Для досягнення мети були поставлені та вирішені наступні завдання:

- провести дослідження сучасних технологій розробки соціальних мереж;
- проаналізувати вже існуючі соціальні мережі;
- розробити архітектуру системи, включаючи функціонал і інтерфейс;
- розробити заходи для покращення безпеки та приватності користувачів;
- створити зручний інтерфейс для користувачів та розробити можливості персоналізації та інтерактивного спілкування;
- протестувати розроблений веб-додаток.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Виявлення загальних переваг та недоліків соціальних мереж

Наше існування сьогодні неможливо уявити без активного онлайн-спілкування. Раніше, ми очікували листи на особисту пошту або на телефонний дзвінок, але сьогодні наш день не розпочнеться, поки ми не перевіримо наявність повідомлень або не переглянемо актуальні новини у соціальних мережах. Ми також не можемо без постійних перевірок повідомлень у наших месенджерах, таких як Telegram, Viber, чи WhatsApp. Зараз, кожної секунди, мільйони користувачів активно взаємодіють один з одним у соціальних мережах.

Метою даної АВР є створення веб-додатку соціальної мережі. Ця система буде складатися з серверної частини та інтерфейсу веб-додатку, який буде представлений у вигляді веб-сторінок.

Найважливіше – визначити сутність соціальної мережі, її переваги та недоліки.

Соціальна мережа – це веб-платформа або програмне забезпечення, яке дозволяє користувачам створювати особисті профілі, обмінюватись інформацією, спілкуватися з іншими користувачами та будувати віртуальні зв'язки[1].

Такі мережі дозволяють людям знаходити старих друзів, знайомитись з новими людьми, обмінюватись думками та інформацією, а також виражати свої інтереси та погляди.

Перші соціальні мережі з'явилися на початку 21 століття, з появою таких платформ як «SixDegrees» в 1997 році. Справжній бум соціальних мереж прийшов у 2000-х роках, і їх популярність швидко зросла завдяки платформам, таким як Facebook, LinkedIn, Twitter та багатьом іншим.

Як і всі веб-додатки соціальні мережі мають численні переваги та недоліки. Розглянемо деякі з них.

Серед переваг соціальних мереж можна виділити такі основні аспекти:

– **Спілкування:** Вони надають можливість взаємодіяти зі знайомими та незнайомими людьми через повідомлення, коментарі, чати тощо.

– **Знаходження інформації:** Користувачі можуть отримувати новини, інформацію та оновлення від різних джерел та поділитися ними з іншими.

– **Бізнес та маркетинг:** Багато компаній використовують соціальні мережі для спілкування з клієнтами, реклами своїх товарів та послуг, а також зворотного зв'язку.

– **Самовираження:** Соціальні мережі надають можливість людям виражати свої думки, ідеї та творчість через текст, фотографії, відео та інші медіа.

– **Розваги та дозвілля:** Соціальні мережі також мають розважальний аспект, який дозволяє користувачам дивитися відеоролики, грати в ігри, слухати музику та інші види розваг та дозвілля.

Також і серед недоліків соціальних мереж можна виділити такі аспекти як:

– **Приватність та безпека:** Інформація, розміщена на соціальних мережах, може бути доступною для небажаних осіб, існує ризик зловживання особистою інформацією або кібербулінгу.

– **Відповідальність за контент:** Відсутність контролю над контентом, який користувачі публікують, може призвести до поширення неправдивої інформації, фейкових новин та некоректного вмісту.

– **Залежність:** Використання соціальних мереж може стати привадою, що веде до втрати часу та зменшення продуктивності.

– **Спам та реклама:** Спам та нав'язлива реклама можуть заважати користувачам отримувати корисний вміст та спілкуватися зі знайомими.

– **Кіберзлочинність:** Соціальні мережі можуть бути місцем діяльності кіберзлочинців, які вчиняють шахрайські атаки та інші злочини.

Загалом, соціальні мережі мають як позитивні, так і негативні аспекти, і їх вплив на життя суспільства продовжує розвиватись. Кожен користувач повинен бути обізнаним та обережним при користуванні соціальними мережами, щоб максимізувати їх переваги та зменшити недоліки.

Соціальні мережі можна поділити за такими критеріями як аудиторія, призначенням, та платформою[2]. Ось приклади соціальних мереж за цими критеріями:

Аудиторія:

– Facebook: Глобальна соціальна мережа з різноманітною аудиторією, але яка з самого початку розроблялась як соціальна мережа для студентів, наразі вона включає в себе користувачів різних вікових груп та інтересів.

– LinkedIn: Соціальна мережа для професіоналів та підприємців, спрямована на професійні контакти та кар'єрний розвиток.

– TikTok: Популярна соціальна мережа серед молоді, де створений акцент на короткі відеороликах.

Призначення:

– Twitter: Спеціалізується на коротких повідомленнях, інформаційних оновленнях та мікроблогах.

– Instagram: Фокусується на візуальному вмісті, особливо фотографіях та коротких відеороликах.

– Pinterest: Орієнтований на поділ та зберігання графічного вмісту, особливо для інтересів, пов'язаних з дизайном, рецептами.

Платформа:

– YouTube: Відомий відеохостинг, де користувачі завантажують і переглядають відеоролики.

– WhatsApp: Популярний месенджер для текстового спілкування та відео дзвінків.

– Snapchat: Спеціалізується на коротких «сторіс» та повідомленнях з обмеженим доступом.

Ці приклади ілюструють різноманітність соціальних мереж, які відповідають різними потребам і інтересам користувачів.

1.2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Більшість сучасних та популярних соціальних мереж є міжнародними, тобто підтримують різні мови. Головною їх метою є сприяти спілкуванню користувачів, незалежно від географічних, релігійних, політичних або інших обмежень.

На початку 2023 року кількість користувачів соціальних мереж становила 4,76 мільярда осіб, що складає приблизно 60% від загального населення світу. Очікується подальший ріст цих показників завдяки розповсюдженню мобільних пристроїв та мобільних соціальних мереж на ринках, які раніше були менше розвинутими.

Розглянемо та проаналізуємо найпопулярніші соціальні мережі які існують на сьогоднішній день[3].

1.2.1 INSTAGRAM

Instagram – це популярна соціальна мережа, яка спеціалізується на обміні фотографіями та короткими відеозаписами. Основний функціонал Instagram включає в себе можливість завантаження фотографій та відео, їх обробку за допомогою різних фільтрів та інструментів, а також спільне спілкування та обмін враженнями[4].

Instagram був заснований Майклом Крігером і Кевіном Сістромом у жовтні 2010 року. Перша версія додатку була доступна лише на iOS-платформі. За короткий час після запуску Instagram завоював популярність завдяки своєму простому та зручному інтерфейсу. У 2012 році Instagram був придбаний компанією Facebook за \$1 мільярд, що підсилило його статус однієї з найважливіших соціальних мереж у світі.

Для використання функціоналу Instagram користувач повинен зареєструватися та завантажити мобільний додаток на свій смартфон або зайти в веб-додаток через будь який браузер.

Приклад сторінки профілю Instagram представлено на рис.1.1.

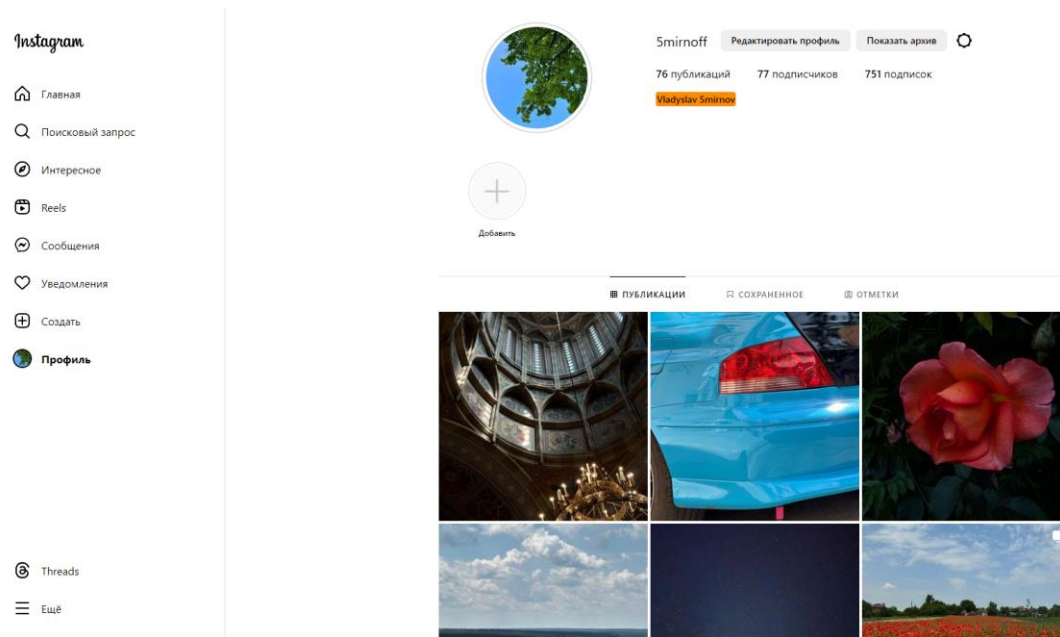


Рисунок 1.1 Сторінка профілю Instagram

Функціонал Instagram включає в себе:

– **Завантаження фотографій та відео:** Користувачі можуть легко додавати свої зображення та короткі відеоролики зі смартфонів.

– **Фільтри та редагування:** Instagram надає широкий вибір фільтрів та інструментів для обробки фотографій та відео, що дозволяє створити креативні зображення.

– **Спільноти та підписки:** Користувачі можуть створювати публічні або приватні профілі, підписуватися на інших користувачів і обмінюватися контентом або власними думками.

– **Історії:** Функція "Історії" дозволяє користувачам додавати тимчасові фотографії та відеозаписи, які автоматично видаляються через 24 години.

– **Живі відеотрансляції:** Користувачі можуть проводити живі трансляції та спілкуватися зі своєю аудиторією в реальному часі.

Як і будь яка соціальна мережа Instagram має свої переваги та недоліки.

Серед переваг можна виділити такі аспекти як:

– **Зручність спілкування та публікації контенту:** Instagram дозволяє легко поділитися своїми миттєвими враженнями, виразити творчість та спілкуватися зі світом.

– **Інтерактивність та спільноти:** Користувачі можуть взаємодіяти, коментувати, вподобати публікації та знаходити нових друзів або зацікавлених осіб.

Серед недоліків Instagram можна виділити:

– **Споживацька природа:** Instagram може сприяти порівнянню свого життя з життям інших користувачів, що може викликати почуття недостатньої самооцінки.

– **Проблеми з конфіденційністю та приватністю:** Публічні профілі можуть використовуватися для збирання даних маркетинговими компаніями та рекламодавцями.

З точки зору програміста я для себе можу також виділити як і переваги Instagram так і його недоліки.

Переваги Instagram з точки зору програміста:

– **Доступність API:** Instagram надає доступ до свого API, що дозволяє розробникам створювати додатки та сервіси, які інтегруються з платформою. Це відкриває можливості для розширення функціоналу та створення додатків, які взаємодіють з Instagram.

– **SDK для мобільних платформ:** Instagram пропонує SDK (набір інструментів розробки програмного забезпечення) для розробки мобільних додатків для платформ iOS та Android. Це спрощує розробку додатків, які взаємодіють з Instagram на мобільних пристроях.

– **Можливість створення бізнес-рішень:** Для програмістів, які спеціалізуються на розробці бізнес-застосунків, Instagram надає можливості для створення інтегрованих рішень, спрямованих на підтримку маркетингу та комунікації брендів.

– **Інновації і оновлення:** Instagram постійно впроваджує нові функції та можливості, що створює простір для творчості та розробки нових додатків, які використовують ці інновації.

Недоліки Instagram з точки зору програміста:

– **Обмеження API:** Instagram обмежує доступ до певних функцій та даних через своє API. Це може обмежити можливості розробників у створенні додатків і сервісів.

– **Зміни в політиці конфіденційності та обмеження:** Політика конфіденційності та правила використання API можуть змінюватися, що вимагає від розробників постійно слідкувати за оновленнями та адаптувати свої додатки до нових вимог.

– **Конкуренція:** Instagram – це дуже конкурентний ринок, і для розробників може бути складно виділитися серед численних додатків та сервісів, які взаємодіють з платформою.

– **Залежність від змін у політиці Instagram:** Розробники можуть бути вразливі до змін у політиці конфіденційності та обмежень, оскільки їх додатки можуть бути піддані впливу з боку Instagram.

Отже, у висновку, Instagram є однією з найпопулярніших та важливих соціальних мереж у світі, яка грає значущу роль у спілкуванні та обміні візуальним контентом для мільярдів користувачів. Історія створення та розвитку Instagram свідчить про його вплив та значення в цьому просторі.

Для користувачів Instagram надає можливість легко ділитися своїми миттєвими враженнями, виразити творчість та спілкуватися зі світом через фотографії та короткі відеоролики. Взаємодія та підписки дозволяють створювати інтерактивні спільноти та знаходити нових друзів або зацікавлених осіб.

Проте Instagram також має свої недоліки, включаючи споживацьку природу, яка може сприяти порівнянню з іншими користувачами і викликати почуття недостатньої самооцінки. Також існують питання конфіденційності та приватності, особливо для публічних профілів.

З точки зору програмістів, Instagram пропонує доступ до API та SDK для розробки додатків, що інтегруються з платформою. Це відкриває можливості для створення додатків, які розширюють функціонал і сприяють інноваціям у

використанні візуального контенту. Однак, існують обмеження API та потреба в постійному оновленні додатків під зміни в політиці та обмеженнях.

У будь-якому випадку, Instagram продовжує відігравати важливу роль у способі, яким ми спілкуємося та виражаємо себе в онлайн-середовищі, надаючи платформу для креативності, спілкування та взаємодії зі світом.

1.2.2 FACEBOOK

Facebook - це одна з найпопулярніших соціальних мереж у світі, яка дозволяє користувачам спілкуватися, обмінюватися відомостями та контентом, створювати профілі та спільноти і багато іншого.

Facebook був заснований Марком Цукербергом, Едуардо Саверіном, Дастіном Московіцом та Крісом Х'юзом у 2004 році під назвою "TheFacebook" на початку як соціальна мережа для студентів Гарвардського університету. Поступово платформа розширилася на інші університети та стала доступною всім користувачам з електронною адресою. У 2012 році Facebook провів успішну ініціальну публічну пропозицію акцій і вступив до фондового ринку NASDAQ[5].

Для користування Facebook користувачу також потрібно зареєструвати особистий профіль. Приклад головної сторінки веб-додатку Facebook представлено на рис.1.2.

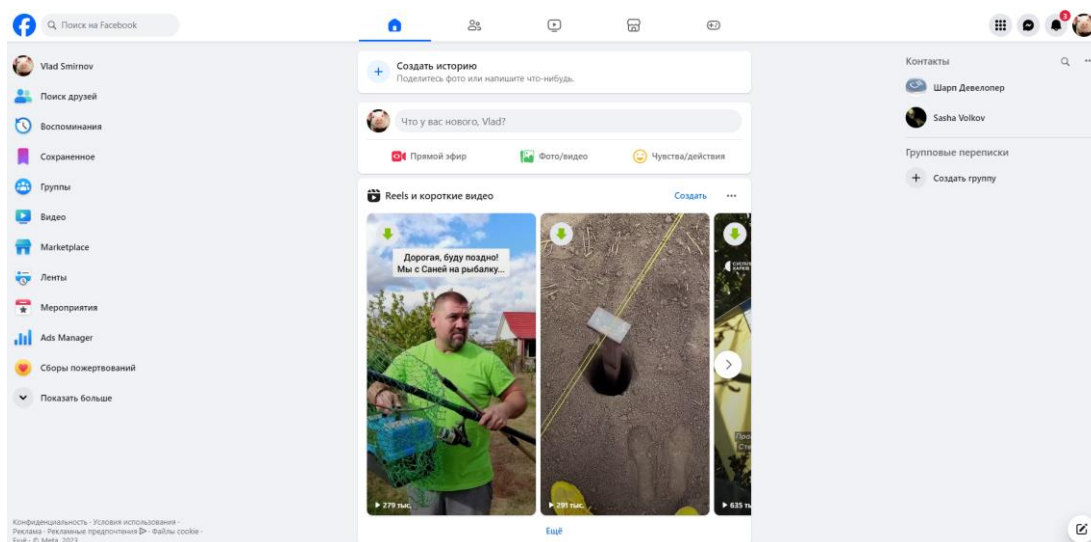


Рисунок 1.2 Головна сторінка Facebook

Функції Facebook включають в себе[6]:

– **Стіну та профілі:** Користувачі можуть створювати власні профілі, додавати друзів, публікувати повідомлення на стіні та обмінюватися коментарями.

– **Месенджер:** Facebook Messenger – це додаток для обміну повідомленнями, відео та аудіо дзвінками.

– **Групи та спільноти:** Користувачі можуть створювати групи та приєднуватися до спільнот зі спільними інтересами.

– **Події:** Можливість створювати та приєднуватися до подій, включаючи віртуальні заходи та зустрічі.

Як і будь яка соціальна мережа Facebook має свої переваги та недоліки.

Серед переваг можна виділити такі аспекти як:

– **Спілкування і зв'язок:** Facebook дозволяє зберігати зв'язок з друзями та родиною, а також знайомитися з новими людьми.

– **Велика аудиторія:** З більш ніж 2 мільярдами активних користувачів, Facebook надає доступ до широкої аудиторії.

– **Різноманітність функцій:** Платформа пропонує різноманітні можливості, включаючи спільноти, події та інше.

– Серед недоліків можна виділити такі аспекти як:

– **Приватність і безпека:** Проблеми з конфіденційністю та збором даних були публічно відомі, що створило обурення серед деяких користувачів.

– **Відволікаючі вміст і реклама:** На Facebook може бути багато відволікаючого контенту та реклами.

З точки зору програміста я для себе можу також виділити як і переваги Facebook так і його недоліки.

Переваги Facebook з точки зору програміста:

– **Розширені можливості API:** Facebook надає багато API, які дозволяють розробникам створювати додатки, які взаємодіють з платформою.

– **Рекламні можливості:** Для розробників і компаній доступні рекламні інструменти, які дозволяють досягти цільової аудиторії.

Недоліки Instagram з точки зору програміста:

– **Специфічні обмеження API:** Facebook має деякі обмеження API, які можуть бути обмежуючими для деяких додатків.

– **Зміна політики та правил:** Політика Facebook може змінюватися, що вимагає постійного оновлення додатків та сервісів

Отже, у висновку, Facebook є однією з найбільших та найважливіших соціальних мереж у світі, яка відіграє важливу роль у спілкуванні та обміні інформацією для мільярдів користувачів. Історія створення та розвитку Facebook свідчить про його вплив та значущість у цьому просторі.

Для користувачів Facebook надає можливості спілкування, створення власних профілів, приєднання до спільнот та участь у подіях. Для розробників Facebook пропонує розширені можливості API та інструменти для реклами, що робить його привабливою платформою для розробки додатків та рекламних рішень.

Проте Facebook також має свої недоліки, зокрема, питання приватності та безпеки даних, а також відволікаючий контент та рекламу. Розробники можуть стикатися зі специфічними обмеженнями API та змінами в політиці, що може потребувати постійного оновлення їх додатків.

У будь-якому випадку, Facebook продовжує залишати важливий вплив на спосіб, яким ми спілкуємося та взаємодіємо в онлайн-середовищі.

1.2.3 TWITTER(X)

Twitter – це соціальна мережа та онлайн-платформа для мікроблогінгу, яка дозволяє користувачам публікувати короткі повідомлення, відомі як "твіти." В цих повідомленнях, обмежених 280 символами (починаючи з 2017 року, попередньо було 140 символів), користувачі можуть ділитися новинами, враженнями, посиланнями та іншим контентом. Твіти можуть бути публічними або приватними, і користувачі можуть підписуватися на інших користувачів, отримувати їхні оновлення та залишати коментарі[7].

Twitter був запущений в березні 2006 року Еваном Вільямсом, Джеком Дорсі, Ноа Гласом і Бізом Стоуном. Початково платформа створювалася як проект подібний до коротких SMS-повідомлень. За роки свого існування Twitter пережив численні зміни та розширення функціоналу, включаючи додавання зображень, відео, опитувань і функції "Twitter Moments," яка дозволяє користувачам створювати історії на основі твітів.

В 2022 році Twitter придбав Ілон Маск, в 2023 році він перейменував його в «X».

Для користування Twitter користувач повинен обов'язково зареєструватись. Приклад головної сторінки веб-додатку Twitter представлено на рис.1.3

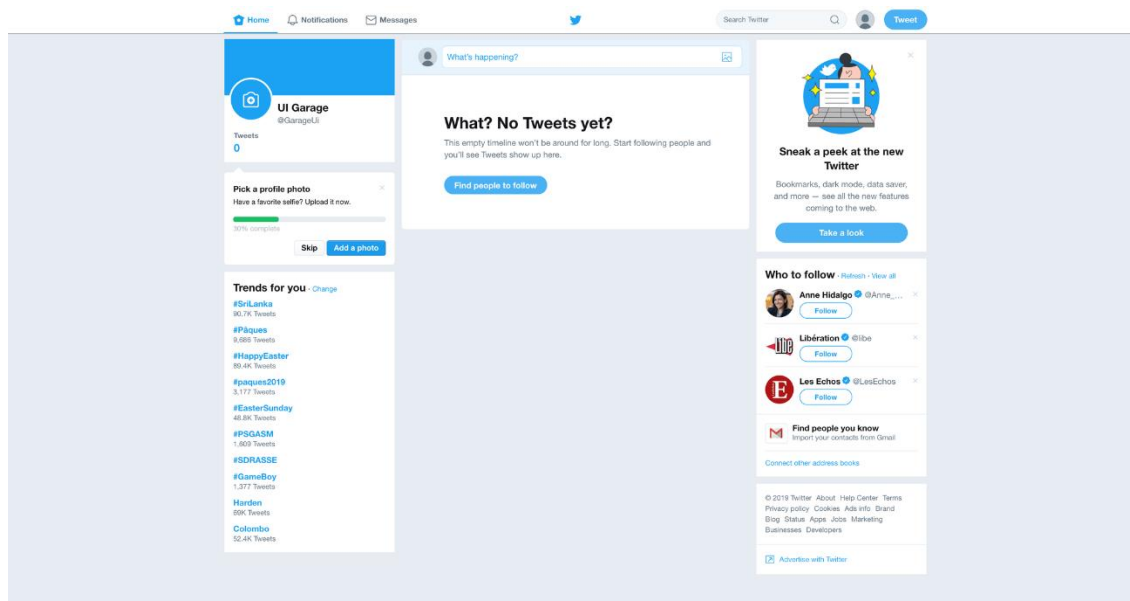


Рисунок 1.3 Головна сторінка Twitter

Основні функції Twitter включають в себе:

– **Публікація твітів:** Twitter дозволяє користувачам публікувати короткі повідомлення, які можуть містити текст, фотографії, відео, посилання та інші медіа-контенти.

– **Підписки:** Користувачі можуть підписуватися на інших користувачів, щоб отримувати їхні твіти в своїй стрічці.

– **Моменти:** Функція "Twitter Moments" дозволяє створювати смислові історії, об'єднуючи твіти та медіаконтент в одному місці.

– **Повідомлення:** Крім публічних твітів, користувачі можуть обмінюватися приватними повідомленнями через "Direct Messages" (DMs).

Як і будь яка соціальна мережа та як вище перераховані соціальні мережі Twitter має свої переваги та недоліки.

Серед переваг можна виділити:

– **Швидкість та актуальність:** Twitter дозволяє користувачам отримувати миттєві оновлення та новини з усього світу.

– **Спілкування:** Взаємодія зі знаменитостями, журналістами та іншими користувачами на публічних платформах.

– **Підтримка спільнот:** Можливість знаходити та приєднуватися до груп і спільнот зі спільними інтересами.

Серед недоліків можна виділити:

– **Черговість інформації:** Завантажена стрічка твітів може бути заплутаною та важкою для відслідковування.

– **Обмеження на кількість символів:** 280 символів може бути недостатньо для повного вираження ідей.

– **Відсутність конфіденційності:** Багато інформації на Twitter є публічною, що може створювати проблеми з приватністю.

Як і в вищезгаданих двох соціальних мережах з точки зору програміста можна виділити схожі переваги та недоліки.

Переваги Twitter з точки зору програміста:

– **API доступність:** Twitter надає доступ до свого API, що дозволяє розробникам створювати додатки та сервіси, які взаємодіють з платформою Twitter.

Недоліки Twitter з точки зору програміста:

– **Обмеження API:** Обмеження щодо кількості запитів та доступу до деяких функцій Twitter API.

– **Складність аналітики:** Аналіз даних Twitter може бути складним завданням через велику кількість інформації та потребу у дорогих інструментах аналізу.

Отже, у висновку, Twitter є однією з найважливіших платформ для мікроблогінгу та спілкування в онлайн-середовищі. Історія його створення свідчить про те, як ця платформа швидко завоювала популярність та стала невід'ємною частиною сучасного цифрового спілкування.

Для користувачів Твіттер надає можливість спілкуватися та обмінюватися інформацією у вигляді коротких твітів, створювати мережу контактів і слідкувати за цікавими подіями та особами. Взаємодія з платформою може бути інтерактивною та динамічною.

З точки зору програмістів Твіттер також має свої переваги, зокрема доступність API, що відкриває можливості для створення додатків, які інтегруються з платформою. Проте, схоже, що наявні обмеження та зміни в політиці використання API можуть становити виклики для розробників.

У всіх великих соціальних мережах, включаючи Твіттер, виникають питання щодо приватності та безпеки даних, а також інформаційного перевантаження від великого потоку інформації та реклами. Розробники та користувачі повинні бути уважними до цих аспектів.

Загалом, Твіттер продовжує відігравати важливу роль у світі спілкування та інформаційного обміну, а його вплив на цей сферу залишається значущим.

1.3 ПОРІВНЯННЯ ІСНУЮЧИХ РІШЕНЬ З РОЗРОБЛЮВАНОЮ СИСТЕМОЮ

Після ознайомлення та аналізу провідних фігур у даній галузі(соціальних мереж і веб-додатків), можна прийти до висновку що цей напрямок розробки справді багато обіцяючий. Саме тому темою даної АВР була обрана розробка веб-додатку соціальної мережі.

Отже, після проведення аналізу вже існуючих рішень можна порівняти розроблювану систему з вже існуючими по функціоналу.

Таблиця 1.1 – Порівняння існуючих та розроблюваної системи

	Instagram	Facebook	Twitter	Smirnov Network
Реєстрація, вхід та вихід	+	+	+	+
Валідація введених даних	+	+	+	+
Перевірка на роботу	+	+	+	-
Створення, редагування, видалення дописів	+	+	+	+
Підписка та відписка	+	+	+	+
Чатинг між користувачами	+	+	+	+
Аудіо та відео дзінки	+	+	-	+
Лайки на дописи	+	+	+	+
Коментарі	+	+	+	+
Перегляд профілю	+	+	+	+

Редагування профілю	+	+	+	+
Мобільний додаток	+	+	+	-

Враховуючи порівняльну таблицю, можна зробити висновки, що розроблюваний веб-додаток соціальної мережі володіє майже всіма існуючими функціями в інших популярних соціальних мережах, а деякі соціальні мережі навіть перевершує по функціоналу. В майбутньому розроблювана система може навіть перевершити конкурентів завдяки майбутнім розробкам та ідеям.

1.4 ПОСТАНОВКА ЗАДАЧІ

В результаті виконання аналізу предметної області та аналізу готових рішень сформульовано мету та постановку задачі.

Метою даної АВР є розробка та впровадження веб-додатку соціальної мережі з метою створення платформи, що дасть користувачам можливість спільного обміну інформацією, спілкування, створення та спільного використання контенту.

Необхідно розробити веб-додаток соціальної мережі з наступною функціональністю:

- створення дописів, їх редагування та видалення;
- пошук користувачів;
- перегляд дописів користувачів, їх коментування та лайки;
- рекомендації для користувачів;
- профіль користувача, та можливість його редагування;
- створення системи підписок між користувачами;
- створення онлайн чату в реальному часі між користувачами;
- створення можливості аудіо та відео дзвінків між користувачами;
- створення системи сповіщень.

ВИСНОВОК ДО РОЗДІЛУ 1

У першому розділі АВР проведено важливий та комплексний аналіз предметної області, пов'язаної з соціальними мережами. В результаті досліджень було зроблено кілька ключових висновків, які визначають напрямок подальших досліджень та розробки.

По-перше, відзначено, що соціальні мережі стали неодмінною частиною сучасного інформаційного простору та спільнот, розглянуте саме поняття «соціальна мережа». Вони забезпечують ефективний засіб комунікації, обміну інформацією та підтримки відносин між користувачами, що робить їх важливими для широкого кола сфер життя, включаючи особисте, професійне та бізнесове оточення.

По-друге, були розглянуті переваги та недоліки соціальних мереж. Переваги включають легкий доступ до інформації, можливість побудувати мережу контактів, підвищити свою видимість та ефективність комунікації. Недоліки включають проблеми з приватністю та безпекою даних, можливість виникнення конфліктів та залежності від соціальних мереж.

По-третє, проведено аналіз існуючих рішень у сфері соціальних мереж та розроблено порівняльну таблицю, яка допоможе визначити переваги розроблюваної соціальної мережі порівняно з існуючими аналогами.

По-четверте, була визначена мета даної АВР та задачі які перед нею стоять.

Зазначені висновки і результати аналізу представляють важливий внесок у подальший розвиток АВР та допоможуть визначити напрямок подальших досліджень та дій у реалізації проекту соціальної мережі.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 ВИЗНАЧЕННЯ ЦІЛЕЙ, ДЕРЕВО ЦІЛЕЙ.

Для визначення основних цілей розробки веб-додатку соціальної мережі потрібно побудувати дерево цілей.

Дерево цілей - це інструмент, який використовується для систематизації та ієрархічного представлення конкретних цілей та підцілей в рамках більшої стратегічної цілі або завдання. Основною ідеєю дерева цілей є розчленування загальної мети на менші, конкретні цілі та завдання, що допомагає зрозуміти і керувати процесом досягнення цілі[8].

Основне правило побудови дерева цілей - принцип ієрархічної структури. Дерево цілей складається з кореневої цілі, яка розчленовується на підцілі, які, в свою чергу, можуть розчленовуватися на ще більш конкретні підцілі і так далі. Найбільш загальна ціль розташована вище, а найбільш конкретні - нижче.

Побудова дерева цілей допомагає керівникам та виконавцям проектів чітко уявити послідовність дій, необхідних для досягнення більшої цілі, та забезпечити легкість визначення пріоритетів і відслідковування прогресу. Цей інструмент широко використовується в управлінні проектами, стратегічному плануванні та у багатьох інших галузях для кращого управління та досягнення поставлених завдань[9].

Розглянемо дерево цілей, яких необхідно досягти, для успішної реалізації проекту, представлено на рис 2.1.

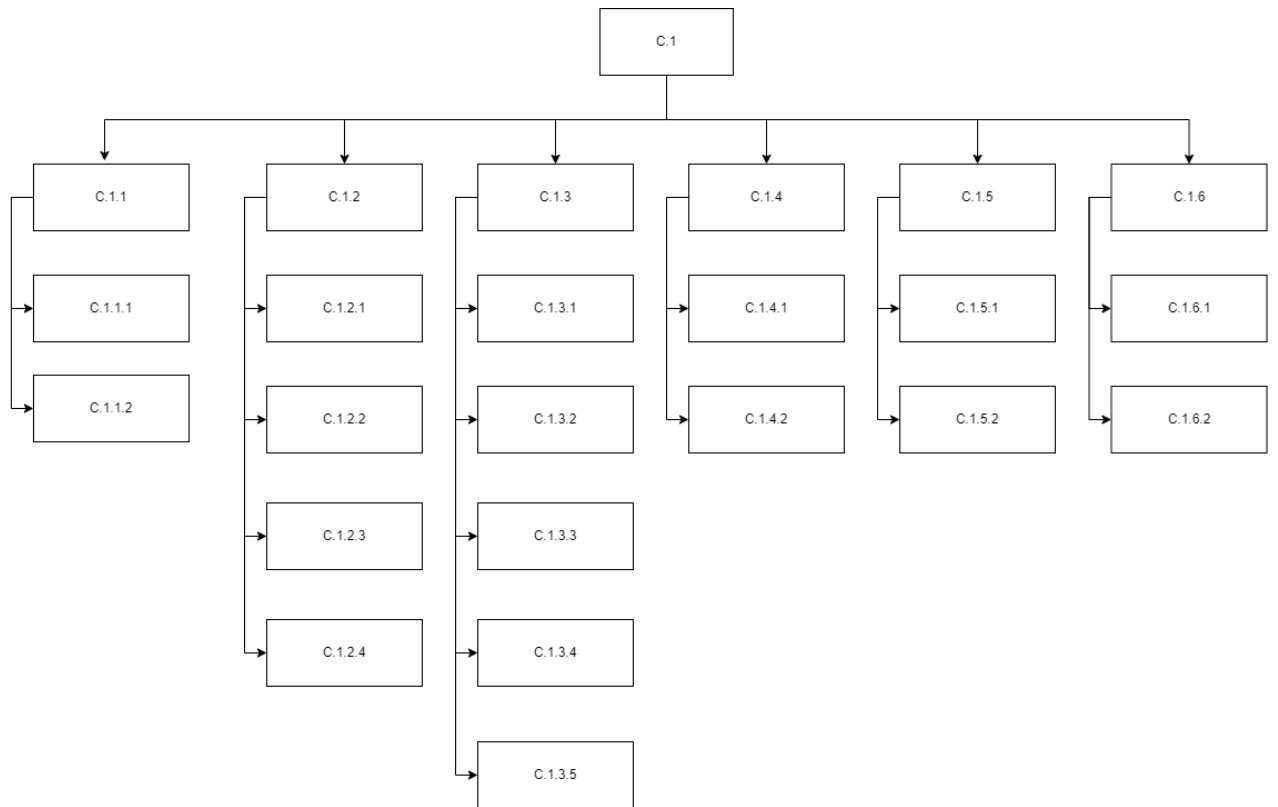


Рисунок 2.1 Дерево цілей розробки веб-додатку соціальної мережі

Детальний опис дерева цілей розробки веб-додатку соціальної мережі:

C.1 – Розробка веб-додатку соціальної мережі;

C.1.1 – Розробка серверної частини;

C.1.1.1 – Розробка серверної архітектури;

C.1.1.2 – База даних та збереження даних;

C.1.2 – Розробка клієнтського інтерфейсу;

C.1.2.1 – Дизайн інтерфейсу;

C.1.2.2 – Реєстрація та авторизація користувачів;

C.1.2.3 – Публікації та стрічка активностей;

C.1.2.4 – Профіль користувача;

C.1.3 – Базова функціональність соціальної мережі;

C.1.3.1 – Створення профілів користувачів;

C.1.3.2 – Спілкування між користувачами;

C.1.3.3 – Можливість додавання підписок;

- C.1.3.4** – Коментування та вподобання публікацій;
- C.1.3.5** – Створення публікацій;
- C.1.4** – Безпека та конфіденційність;
- C.1.4.1** – Захист від несанкціонованого доступу;
- C.1.4.2** – Конфіденційність особистих даних;
- C.1.5** – Тестування та налагодження;
- C.1.5.1** – Тестування функціональності;
- C.1.5.2** – Виявлення та виправлення помилок;
- C.1.6** – Запуск та розвиток;
- C.1.6.1** – Запуск веб-додатку;
- C.1.6.2** – Підтримка та розвиток функціональності.

2.2 РОЗРОБКА ТЕХНІЧНОГО ЗАВДАННЯ

Розробка будь-якого програмного продукту не можлива без створення ескізного проекту, в якому відображено результати зовнішнього проектування ПЗ.

Для початку проектування веб-додатку соціальної мережі, гарним відправним пунктом може бути DFD(Data Flow Diagram). Проте, через обрану об'єктноорієнтовану методологію, планується перейти до використання мови моделювання UML.

Графічне представлення DFD яскраво демонструє переміщення інформації всередині процесу або системи.

Для ілюстрації вхідних і вихідних даних, об'єктів зберігання і переміщення інформації між джерелами і пунктами доставки в таких діаграмах використовуються стандартні геометричні фігури, такі як прямокутники і кола, а також стрілки та короткі текстові мітки.

DFD можуть варіюватися від простих уявних зображень процесів до складних багаторівневих схем з глибоким аналізом методів обробки даних[10]. Вони використовуються для аналізу та моделювання нових систем. В контексті

найкращих практик візуалізації даних, DFD часто надають візуальну інформацію про процеси, які важко пояснити словами, і сприяють ефективному сприйняттю інформації як для розробників, так і для керівників вищого рівня.

У свою чергу, мова моделювання UML (аббревіатура від "Unified Modeling Language" або "Об'єднана Мова Моделювання") надає можливість детально відобразити проект та досягти необхідного рівня деталізації.

2.2.1 DATA FLOW DIAGRAM

Створення DFD розпочинається з нульового рівня, який є контекстною діаграмою.

Контекстна діаграма - це графічне зображення системи, яке надає загальний опис системи та її взаємодії з зовнішнім середовищем[11].

Контекстна діаграма (DFD 0) веб-додатку соціальної мережі представлена на рис. 2.2.



Рисунок 2.2 Контекстна діаграма веб-додатку соціальної мережі

Для більш детального розгляду веб-додатку побудуємо DFD 1. На цьому рівні показуються основні функції системи та розбивається DFD 0-го на підпроцеси. DFD 1 веб-додатку соціальної мережі представлена на рис. 2.3.

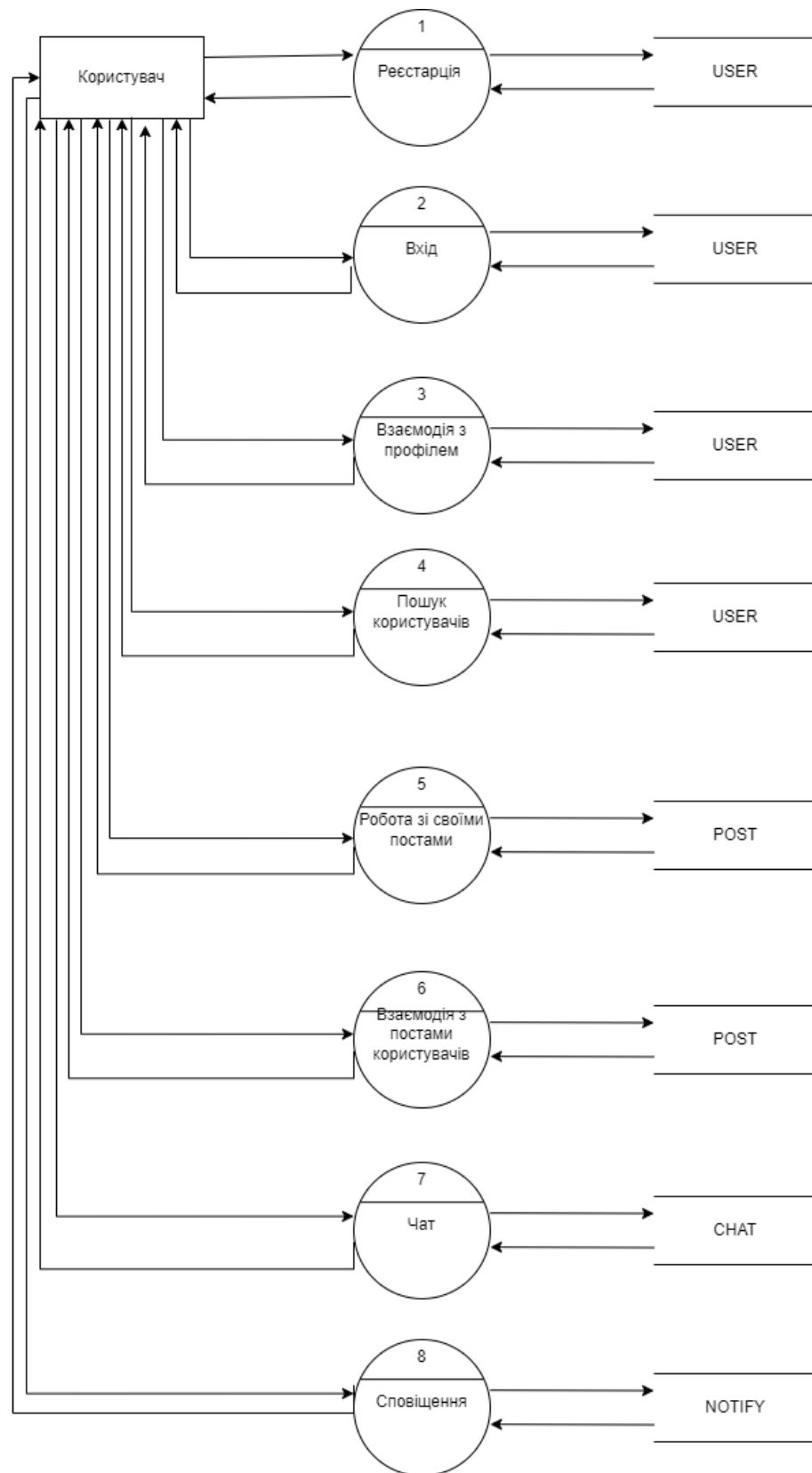


Рисунок 2.3 DFD 1-го рівня веб-додатку соціальної мережі

На діаграмі першого рівня зображено 8 основних процесів веб-додатку соціальної мережі, а саме:

1) **Реестарція** – процес який відповідає за реестрацію користувача;

2) **Вхід** – процес який відповідає за вхід користувача на сайт та його аутентифікацію;

3) **Взаємодія з профілем** – процес який відповідає за взаємодію з власним профілем, його редагування.

4) **Пошук користувачів** – процес який відповідає за пошук, перегляд та взаємодію з зареєстрованими в соціальній мережі користувачами;

5) **Робота зі своїми дописами** – процес який відповідає за створення власного допису, його редагування та видалення;

6) **Взаємодія з дописами користувачів** – процес який відповідає за взаємодію з дописами користувачів, а саме лайки, коментарі, поширення.

7) **Чат** – процес який відповідає за створення чату та спілкування користувачів.

8) **Сповіщення** – процес який відповідає за взаємодію зі сповіщеннями, а саме їх перегляд та видалення.

Для більше детального розгляду системи проведемо декомпозицію процесу «Робота зі своїми дописами» представлена на рис 2.4.

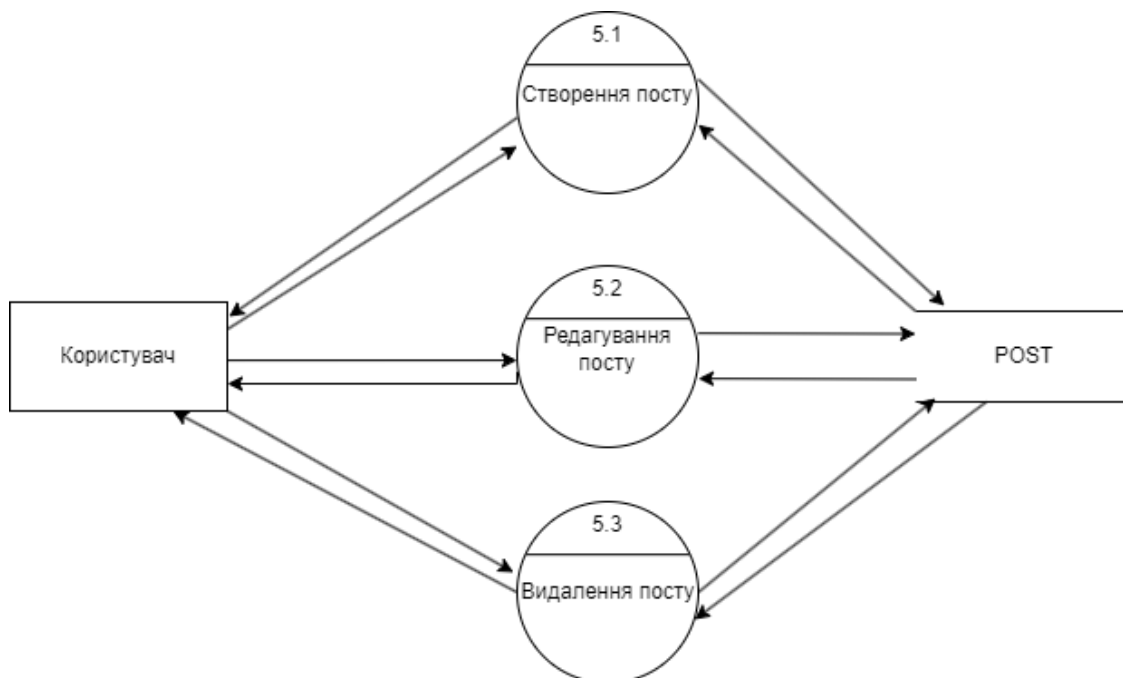


Рисунок 2.4 Декомпозиція процесу робота зі своїми дописами

Отже, була розроблена DFD нульового, першого та другого рівня веб-додатку соціальної мережі.

2.2.2 КОНЦЕПТУАЛЬНА МОДЕЛЬ

Концептуальна модель - це абстрактне представлення структури даних і відносин між ними без урахування технічних аспектів реалізації бази даних.

Концептуальна модель допомагає описати дані та їх відносини на високому рівні абстракції. Вона допомагає розуміти сутність і взаємозв'язки даних без врахування деталей реалізації. Це спрощує розуміння потреб користувачів та бізнес-логіки системи[12].

Концептуальна модель для веб-додатку соціальної мережі буде зображена у вигляді ER-моделі.

ER-модель - це конкретний вид концептуальної моделі, яка використовує сутності (entities) для представлення об'єктів та їх атрибутів, а також зв'язки (relationships) для відображення взаємозв'язків між сутностями[13].

ER-модель допомагає формалізувати структуру даних та взаємозв'язки між ними. Вона дозволяє ідентифікувати ключові сутності, атрибути та зв'язки в системі. ER-модель є важливим інструментом для розробки баз даних, проектування та аналізу взаємозв'язків між даними, а також для спрощення комунікації між розробниками та стейкхолдерами.

Загальна мета концептуальної моделі та ER-моделі полягає в тому, щоб створити зрозуміле та узгоджене представлення даних та їх відносин для подальшого проектування та розробки бази даних. Вони допомагають зменшити ризики помилок і непорозумінь при створенні інформаційних систем, а також полегшують подальшу розробку та підтримку даних.

Концептуальна модель даних представлена у вигляді ER-моделі та представлена на рис. 2.5. Опис зв'язків концептуальної моделі представлений у таблиці 2.1.

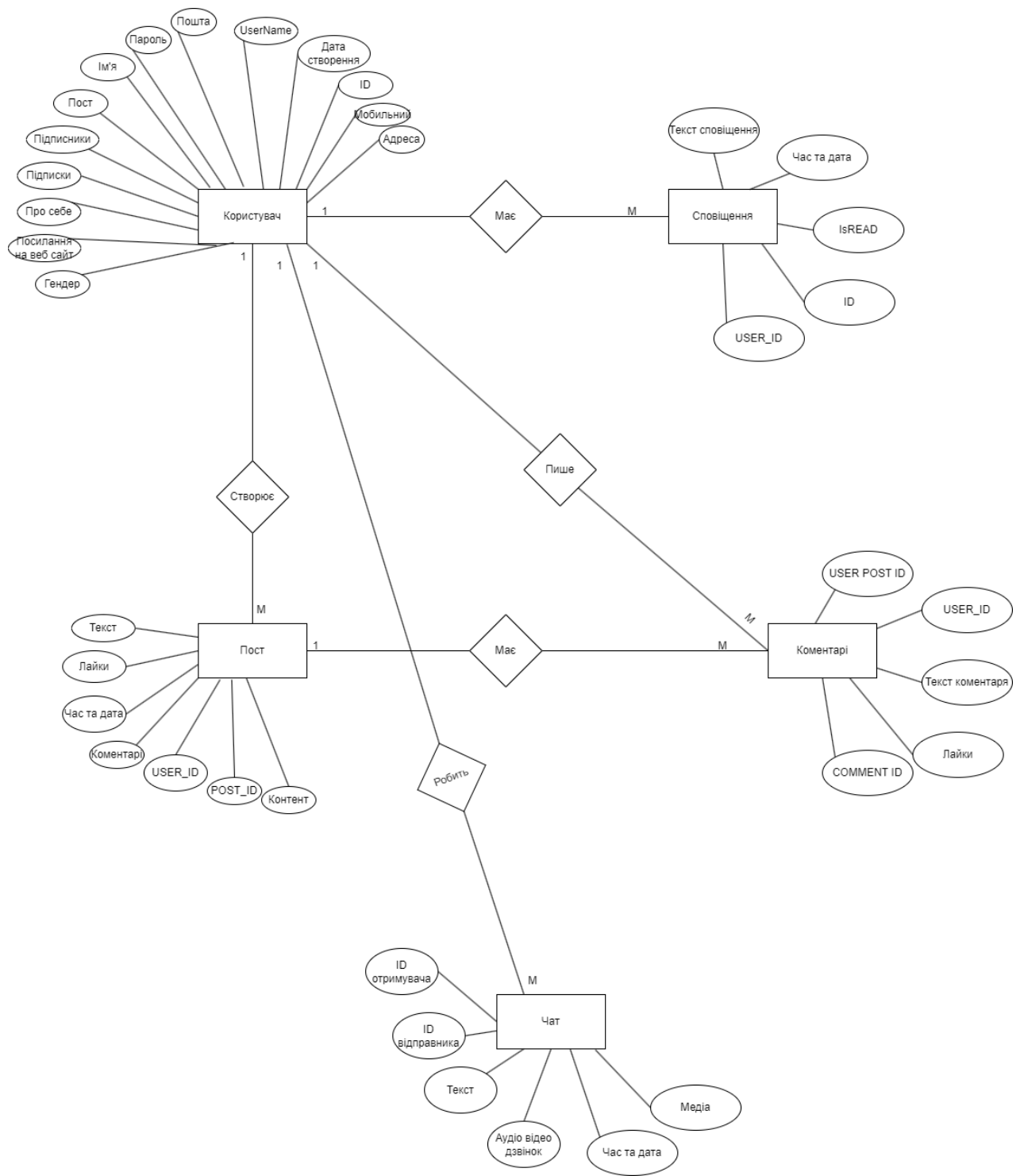


Рисунок 2.5 ER-модель веб-додатку соціальної мережі

Таблиця 2.1 – Опис зв'язків концептуальної моделі

Сутності	Тип зв'язку	Опис
1	2	3
Користувач – сповіщення	1:M	Один користувач може мати безліч сповіщень.
Користувач – допис	1:M	Один користувач може створити безліч дописів.
Користувач – чат	1:M	Один користувач може робити безліч чатів.
Користувач – коментарі	1:M	Один користувач може написати безліч коментарів.
Допис – коментарі	1:M	Один допис може мати безліч коментарів від користувачів

2.3 UML ДІАГРАМИ ПРЕЦЕДЕНТІВ, КЛАСІВ, ПОСЛІДОВНОСТІ, ДІЯЛЬНОСТІ

2.3.1 ДІАГРАМА ПРЕЦЕДЕНТІВ

Діаграма прецедентів (Use Case Diagram) є одним із видів діаграм в мові моделювання Unified Modeling Language (UML), і вона використовується для відображення функціональних вимог до системи або програмного продукту з точки зору користувачів. Діаграма прецедентів допомагає моделювати, як користувачі взаємодіють з системою, які функціональності системи вони використовують та як система взаємодіє з користувачами[14].

Основні елементи, що можуть бути включені до діаграми прецедентів, включають[15]:

- **Прецеденти (Use Cases):** Прецеденти представляють окремі функціональні можливості або послуги, які система надає користувачам. Кожен прецедент має назву та може бути описаний деталізованою документацією.

- **Актори (Actors):** Актори представляють ролі або сутності, які взаємодіють з системою. Актори можуть бути користувачами, зовнішніми системами, іншими програмами або навіть іншими частинами системи.

- **Відношення між акторами та прецедентами:** Діаграма відображає, які актори взаємодіють з якими прецедентами. Це описує, які функції системи доступні для кожного актора.

- **Включення та розширення:** Діаграма може також показувати включення та розширення прецедентів, що описують, які додаткові функції можуть бути включені або розширені в межах інших прецедентів.

Діаграма прецедентів корисна для наступних цілей:

- **Розуміння вимог користувачів:** Вона допомагає розробникам та аналітикам зрозуміти, які функції та послуги очікуються від системи від користувачів.

- **Комунікація зі зацікавленими сторонами:** Діаграма прецедентів слугує ефективним інструментом для спілкування з клієнтами, менеджментом та іншими зацікавленими сторонами для з'ясування та підтвердження вимог.

– **База для подальшого проектування системи:** Вона може бути використана як основа для подальшого проектування архітектури системи та розробки функціональності.

– **Тестування та верифікація:** Вона допомагає розробникам та тестувальникам розуміти, як система повинна вести себе в різних сценаріях взаємодії з користувачами.

Було розроблено діаграму прецедентів для веб-додатку соціальної мережі представлено на рис. 2.6.

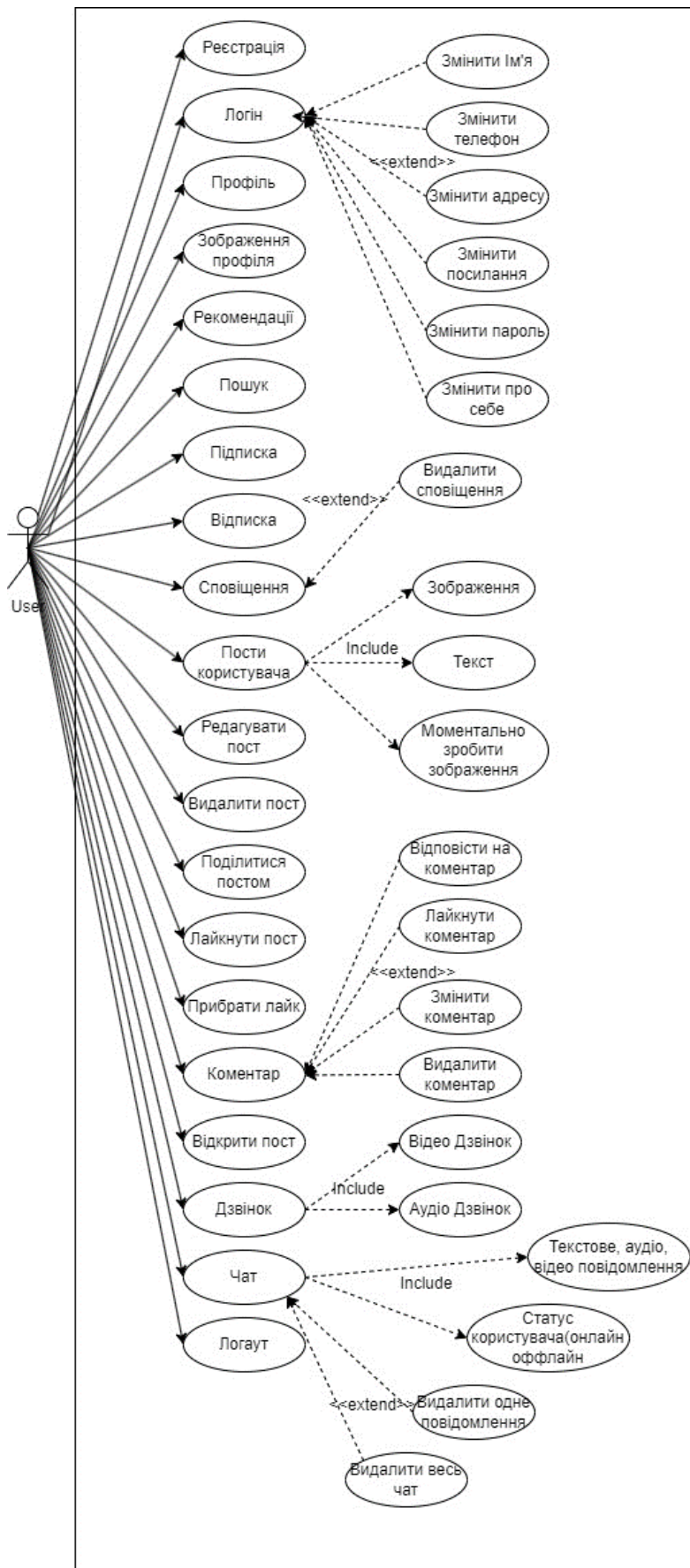


Рисунок 2.6 Діаграма прецедентів

2.3.2 ДІАГРАМА КЛАСІВ

Діаграма класів є одним із видів діаграм в мові моделювання UML (Unified Modeling Language), і вона використовується для моделювання структури об'єктно-орієнтованої системи програмування. Основною метою діаграми класів є відображення класів, їх взаємозв'язків та структури системи. Діаграма класів надає абстрактне представлення об'єктів, які існують у системі, та спільних характеристик цих об'єктів[16].

Основні елементи діаграми класів включають наступне[17]:

– **Класи:** Класи відображають об'єкти або сутності, які існують в системі. Класи містять атрибути (змінні) та методи (функції), які визначають характеристики та поведінку об'єктів.

– **Атрибути:** Атрибути представляють характеристики класу і відображаються в середині класу. Наприклад, якщо є клас "Користувач", то атрибутами можуть бути "ім'я", "прізвище" і т. д.

– **Методи:** Методи визначають поведінку класу і також відображаються всередині класу. Методи описують операції, які можуть бути виконані над об'єктами цього класу.

– **Взаємозв'язки:** Діаграма класів також відображає взаємозв'язки між класами. Це може включати асоціації (зв'язки між об'єктами), спадкування (відносини "батько-потомок"), агрегації (складові частини) та композиції (складові частини зі строгою власністю).

– **Коментарі:** Діаграма класів також може включати коментарі та пояснення для полегшення розуміння моделі.

Основна мета створення діаграми класів це допомогти розробникам та архітекторам систем зрозуміти структуру програми, ідентифікувати класи, їх атрибути та методи, а також визначити, як вони взаємодіють між собою.

Це корисний інструмент для проектування, аналізу та документування програмного коду, що сприяє покращенню розуміння системи та спрощує комунікацію між розробниками.

Отже, було розроблено діаграму класів для веб-додатку соціальної мережі.
 Діаграма класів соціальної мережі представлена на рис. 2.7.

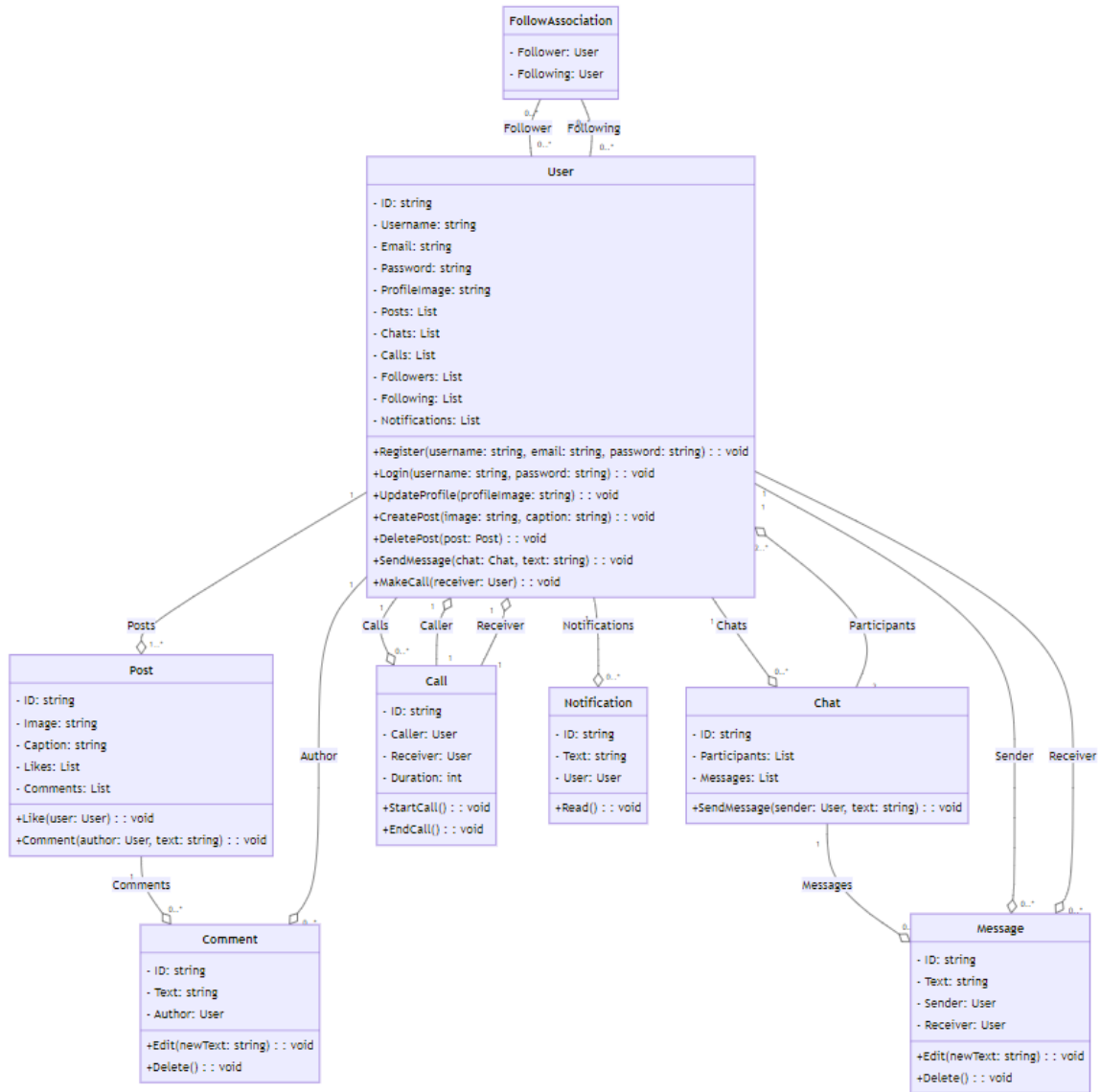


Рисунок 2.7 Діаграма класів

Опишемо поля та методи кожного класу веб-додатку соціальної мережі:

1) User(Користувач):

- Поля:
 - ID: Унікальний індифікатор користувача.
 - Username: Ім'я користувача.
 - Email: Електронна пошта користувача.
 - Password: Пароль користувача.
 - ProfileImage: URL зображення профіля користувача.
 - Posts: Список дописів користувача.
 - Chats: Список чатів, в котрих бере участь користувач.
 - Calls: Список дзвінків, які здійснює користувач.
 - Followers: Список користувачів, підписаних на цього користувача.
 - Following: Список користувачів, на яких підписаний цей користувач.
 - Notifications: Список сповіщень для користувача
- Методи та функції:
 - Register(username, email, password): Реєстрація нового користувача із заданим ім'ям користувача, електронною поштою та паролем.
 - Login(username, password): Аутентифікація користувача із заданим ім'ям користувача та паролем.
 - UpdateProfile(profileImage): Оновлення зображення профілю користувача.
 - CreatePost(image, caption): Створення нового допису із зображенням та описом.
 - DeletePost(post): Видалення допису.
 - SendMessage(chat, text): Надсилання повідомлення в чат.
 - MakeCall(receiver): Дзвінок іншому користувачеві.

2) Post(Допис):

- Поля:
 - ID: Унікальний ідентифікатор допису.
 - Image: URL зображення, прикріпленого до допису.
 - Caption: Текстовий опис допису.
 - Likes: Список користувачів які поставили лайк на цей допис.
 - Comments: Список коментарів до допису.
- Методи та функції:
 - Like(user): Поставити лайк допису.
 - Comment(author, text): Додати коментар до допису.

3) Comment (Коментарі):

- Поля:
 - ID: Унікальний ідентифікатор коментарю.
 - Text: Текст коментарю.
 - Author: Користувач котрий написав коментар.
- Методи та функції:
 - Edit(newText): Редагувати коментар.
 - Delete(): Видалити коментар.

4) Chat(Чат):

- Поля:
 - ID: Унікальний ідентифікатор чату.
 - Participants: Список користувачів які знаходяться в чаті.
 - Messages: Список повідомлень в чаті.
- Методи та функції:
 - SendMessage(sender, text): Відправити повідомлення в чат.

5) Message(Повідомлення):

- Поля:
 - ID: Унікальний ідентифікатор повідомлення.
 - Text: Текст повідомлення.

- Sender: Користувач, що відправив повідомлення.
- Receiver: Користувач, що отримав повідомлення.
- Методи та функції:
 - Edit(newText): Редагувати повідомлення.
 - Delete(): Видалити повідомлення.

6) Call(Дзвінок):

- Поля:
 - ID: Унікальний ідентифікатор повідомлення дзвінка.
 - Caller: Користувач, що телефонує.
 - Receiver: Користувач, що прийняв дзвінок.
 - Duration: Тривалість дзвінка в секундах.
- Методи та функції:
 - StartCall(): Почати дзвінок.
 - EndCall(): Завершити дзвінок.

7) FollowAssociation (Відношення фолловінг/фолловер):

- Поля:
 - Follower: Користувач, котрий підписаний на іншого користувача.
 - Following: Користувач, на якого підписаний Follower.

Для класу FollowAssociation не потрібно методів або функцій, цей клас представляє асоціацію між користувачами, які підписані один на одного.

8) Notification

- Поля:
 - ID: string - унікальний ідентифікатор сповіщення.
 - Text: string - текст сповіщення.
 - User: User - користувач, якому належить сповіщення.
- Методи та функції:
 - Read(): void - позначення сповіщення як прочитаного.

Ці класи, методи та функції представляють собою основну функціональність розроблюваного веб-додатку соціальної мережі, яка дозволяє користувачам створювати дописи, обмінюватись повідомленнями, робити дзвінки та підписуватись на інших користувачів.

2.3.3 ДІАГРАМА ПОСЛІДОВНОСТІ

Діаграма послідовності (Sequence diagram) є одним із видів діаграм в мові моделювання UML (Unified Modeling Language). Ця діаграма використовується для моделювання послідовності взаємодії між різними об'єктами чи компонентами в системі. Діаграма послідовності дозволяє візуалізувати, як об'єкти спілкуються один з одним та в якому порядку ця взаємодія відбувається в ході виконання певної функціональності чи сценарію[18].

Основні елементи діаграми послідовності включають[19]:

- **Об'єкти**(зазвичай представлені вертикальними лініями): Об'єкти представляють різні сутності, які беруть участь у взаємодії. Це може бути користувач, комп'ютер, сервер, модуль програми, тощо.

- **Лінії життя**: Лінії життя вказують на тимчасовий інтервал існування об'єкта під час взаємодії.

- **Послідовність повідомлень (стрілки)**: Стрілки з'єднують об'єкти та вказують, які повідомлення передаються між ними в певній послідовності. Повідомлення можуть бути синхронними (чекають відповіді) або асинхронними (не чекають відповіді).

- **Параметри та повернені значення повідомлень**: Діаграма може включати інформацію про параметри, які передаються в повідомленнях, і значення, які повертаються.

- **Фрагменти операцій**: Діаграма може містити умовні фрагменти, такі як умовні галузі (alt), паралельні фрагменти (par), цикли (loop), тощо, що допомагають відобразити складні сценарії взаємодії.

Діаграми послідовності корисні для розуміння та моделювання процесів взаємодії між об'єктами в системі або між різними компонентами програми. Вони допомагають аналізувати логіку виконання, ідентифікувати потенційні проблеми та оптимізувати послідовність операцій. Діаграми послідовності особливо корисні для розробки та документування сценаріїв взаємодії в програмному коді, системному архітектурному проектуванні та тестуванні.

Отже, була розроблена діаграма послідовностей для веб-додатку соціальної мережі, а саме процес відправки повідомлення представлена на рис. 2.8.

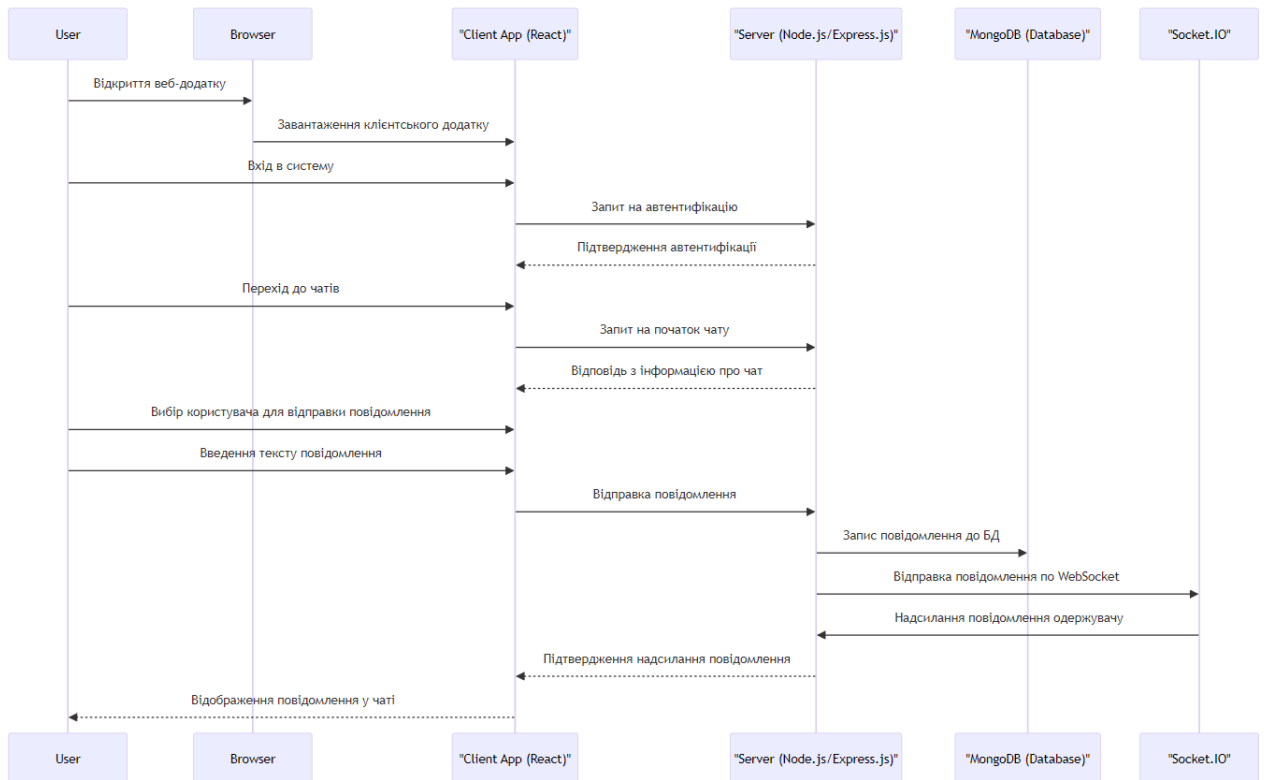


Рисунок 2.8 Діаграма послідовності

Дана діаграма послідовності описує послідовність дій та взаємодії між учасниками (користувачем, браузером, клієнтським додатком, сервером, базою даних та Socket.IO) при відправці повідомлення в веб-додатку соціальної мережі. Давайте розглянемо її покроково:

- **User (Користувач):** Користувач розпочинає взаємодію з додатком.
- **Browser (Браузер):** Браузер користувача завантажує клієнтський додаток (React), який представляє собою інтерфейс соціальної мережі.
- **Client App (React) (Клієнтський додаток):** Користувач увійшов в систему, вводячи свої облікові дані.
- **Server (Node.js/Express.js) (Сервер):** Клієнтський додаток відправляє запит на сервер для аутентифікації користувача.
- **Server (Node.js/Express.js) (Сервер):** Сервер обробляє запит на аутентифікацію та повертає підтвердження користувачеві.

– **Client App (React) (Клієнтський додаток):** Після успішної аутентифікації користувач переходить до чату всередині додатку.

– **Server (Node.js/Express.js) (Сервер):** Клієнтський додаток відправляє запит серверу для початку чату.

– **Server (Node.js/Express.js) (Сервер):** Сервер обробляє запит та відправляє назад інформацію про чат.

– **Client App (React) (Клієнтський додаток):** Користувач вибирає отримувача для відправки повідомлення і вводить текст повідомлення.

– **Client App (React) (Клієнтський додаток):** Клієнтський додаток відправляє повідомлення на сервер.

– **Server (Node.js/Express.js) (Сервер):** Сервер записує повідомлення в базу даних MongoDB.

– **Server (Node.js/Express.js) (Сервер):** Потім сервер використовує Socket.IO для відправлення повідомлення через протокол WebSocket.

– **Socket.IO:** Повідомлення відправляється через WebSocket на сервер Socket.IO.

– **Server (Node.js/Express.js) (Сервер):** Сервер передає повідомлення отримувачу через Socket.IO.

– **Client App (React) (Клієнтський додаток):** Після отримання підтвердження від сервера, клієнтський додаток відображає повідомлення в чаті.

– **User (Користувач):** Користувач бачить повідомлення в чаті.

Отже, була розроблена діаграма послідовності для розроблюваного веб-додатку соціальної мережі, та був розглянутий кожен її крок.

2.3.4 ДІАГРАМА ДІЯЛЬНОСТІ

Діаграма діяльності (Activity Diagram) є одним із типів діаграм в мові моделювання UML (Unified Modeling Language), яка використовується для візуалізації послідовності дій, процесів або діяльностей, які відбуваються в системі. Діаграми діяльності допомагають моделювати різні аспекти функціонування системи, включаючи бізнес-процеси, логіку програмного забезпечення, алгоритми та інші аспекти системи[20].

Основні елементи діаграми діяльності включають в себе[21]:

- **Діяльності (Activities):** Це операції або дії, які виконуються в рамках системи або процесу. Вони представляють кроки або етапи процесу.

- **Рішення (Decisions):** Вказують на рішення, які приймаються на основі певних умов. Вони розділяють потік виконання на різні шляхи залежно від умов.

- **Об'єднання (Merge):** Використовуються для об'єднання різних шляхів виконання після прийняття рішення.

- **Початкові та завершальні точки (Initial and Final Nodes):** Початкові точки позначають початок діаграми, а завершальні точки вказують на завершення діаграми.

- **Флоу (Flow):** Стрілки, які показують послідовність виконання діяльностей та шляхи, по яких виконання може переходити від однієї діяльності до іншої.

- **Об'єднання (Fork):** Використовується для поділу потоку виконання на кілька паралельних потоків.

Діаграми діяльності допомагають розуміти та моделювати логіку процесів або систем, і вони часто використовуються в аналізі та проектуванні програмного забезпечення, а також в бізнес-аналізі для опису бізнес-процесів.

Діаграма діяльності для веб-додатку соціальної мережі представлена на рис. 2.9.

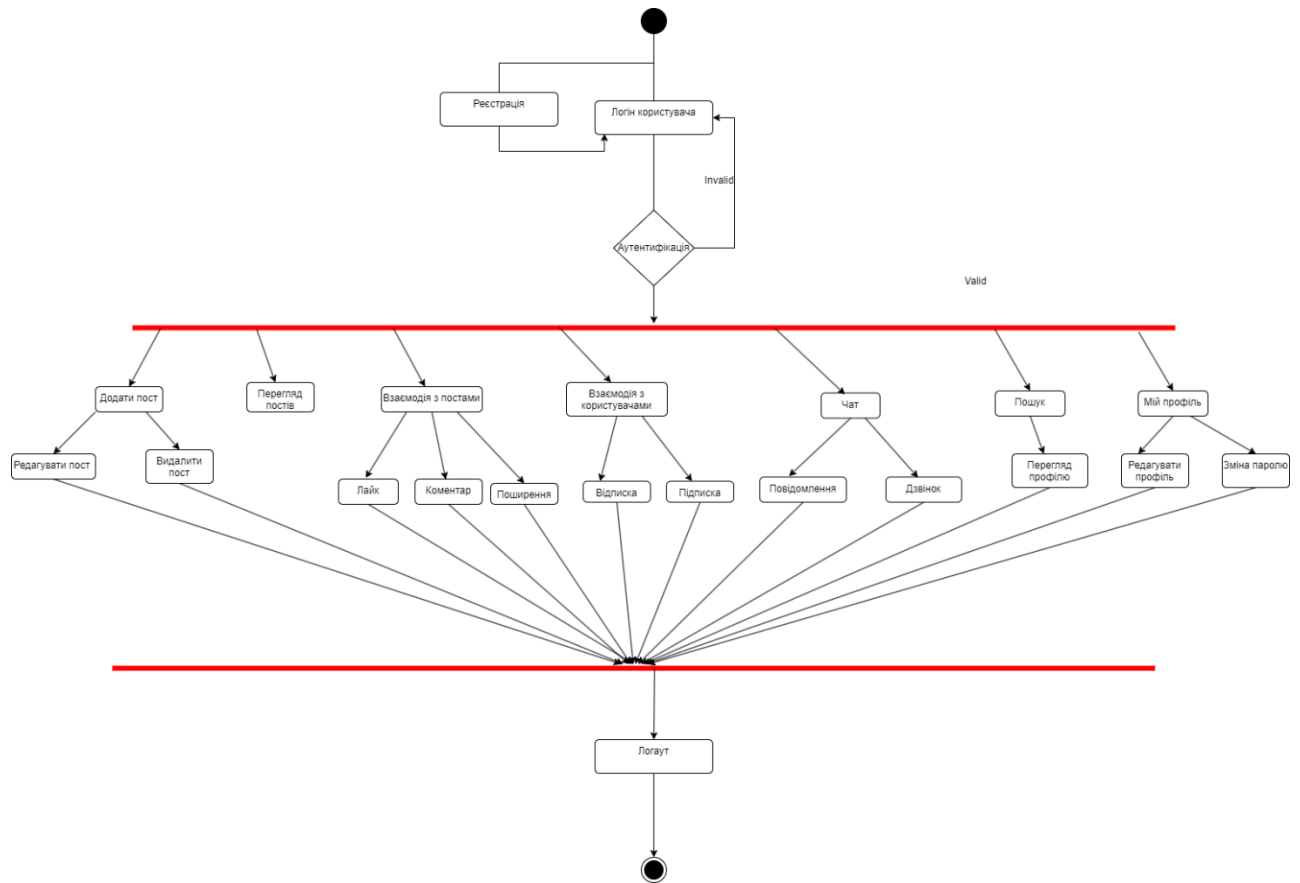


Рисунок 2.9 Діаграма діяльності веб-додатку соціальної мережі

Отже, була розроблена діаграма активності веб-додатку соціальної мережі, на якій зображені основні процеси системи.

ВИСНОВОК ДО РОЗДІЛУ 2

У другому розділі АВР було проведено глибокий аналіз та проектування веб-додатку для соціальної мережі. Описані цілі та завдання розділу були реалізовані через структуровану методологію, включаючи визначення цілей, побудову дерева цілей, розробку DFD діаграми, концептуальної моделі та опис зв'язків в ній.

Особлива увага була приділена розробці UML діаграм, які відображають різні аспекти системи. Діаграма прецедентів послугувала засобом опису взаємодій між користувачами та системою. Діаграма класів надавала зрозумілу структуру об'єктів, їх взаємозв'язки та властивості. Діаграма послідовності визначила порядок виконання операцій та комунікацію між об'єктами. Діаграма діяльності вказувала на послідовність дій у конкретному процесі.

Під час розробки веб-додатку було враховано отримані результати аналізу та проектування, що забезпечило чітку структуру системи та дозволило відобразити функціональність соціальної мережі. Результати цього розділу слугуватимуть основою для подальшого етапу реалізації проекту та допоможуть забезпечити високий рівень функціональності та зручність використання для користувачів соціальної мережі.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-ДОДАТКУ СОЦІАЛЬНОЇ МЕРЕЖІ

3.1 ВИБІР ЗАСОБІВ РОЗРОБКИ

3.1.1 ОГЛЯД СЕРЕДОВИЩА РОЗРОБКИ

Visual Studio Code

В якості середовища для розробки було обрано Visual Studio Code. Розглянемо, що собою представляє це середовище розробки та які переваги воно має.

Visual Studio Code – це безкоштовний відкритий редактор коду, розроблений та підтримуваний компанією Microsoft. Він підтримує роботу з різними мовами програмування і надає розширені можливості редагування, налагодження та керування версіями коду[22].

Логотип Visual Studio Code представлено на рис. 3.1.1.

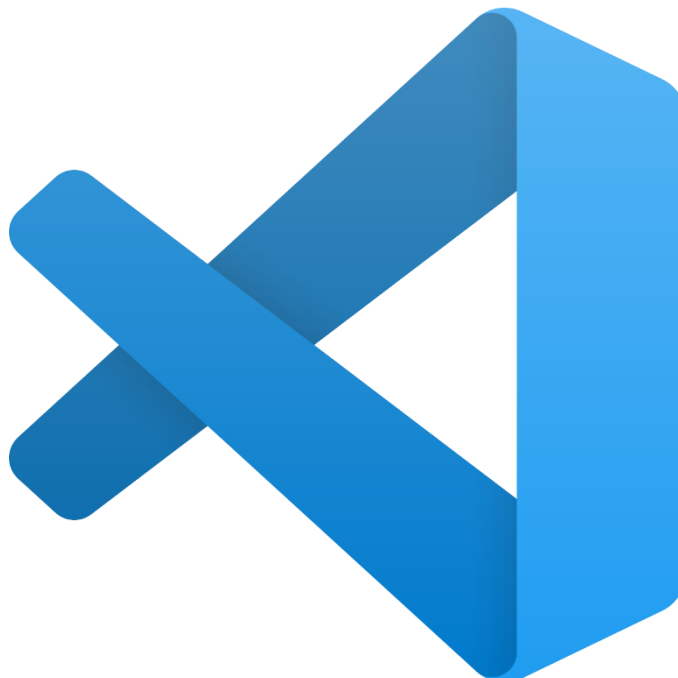


Рисунок 3.1.1 Логотип Visual Studio Code

Серед основних переваг Visual Studio Code можна виділити такі як:

– **Легкість використання:** VS Code володіє інтуїтивно зрозумілим інтерфейсом, що полегшує роботу розробників будь-якого рівня досвіду.

– **Розширюваність:** З великою кількістю розширень та плагінів, ви можете адаптувати VS Code до ваших конкретних потреб та вимог проекту.

– **Підтримка різних мов програмування:** Якщо ваш веб-додаток використовує різні мови програмування для фронтенду та бекенду, VS Code надає підтримку для багатьох з них.

– **Інтеграція з Git:** Вбудована підтримка Git у VS Code дозволяє легко вести роботу з контролем версій і спільно вирішувати задачі в команді.

– **Крос-платформеність:** Можливість використовувати VS Code на різних операційних системах дає свободу вибору платформи для розробки.

– **Активна спільнота та Оновлення:** Спільнота розробників VS Code є активною, а регулярні оновлення додають нові функції та поліпшують продукт.

Отже, основні причини які відіграли роль в виборі VS Code, як середовища для розробки веб-додатку соціальної мережі, це його легкість та зручність використання, розширюваність та підтримки різних мов програмування.

3.1.2 ОГЛЯД ОБРАНИХ МОВ ПРОГРАМУВАННЯ ТА МЕТОДОЛОГІЙ

HTML

Оскільки, HTML є ключовим інструментом для створення веб-сайтів та веб-додатків то і в розробці веб-додатку соціальної мережі він також був використаний. Розглянемо, що представляє собою HTML, та які його переваги.

HTML (Hypertext Markup Language) - це стандартна мова розмітки для створення та відображення веб-сторінок. Вона використовується для структурування змісту веб-сторінки за допомогою елементів, які відомі як "теги". HTML може включати текст, зображення, посилання, форми, відео, аудіо та інші елементи. Кожен тег визначає роль або значення для елемента, який він огортає[23].

Основні характеристики HTML:

- **Теги:** Теги - це основна будівельна одиниця HTML. Вони визначають структуру та семантику веб-сторінки. Наприклад, `<p>` вказує на початок абзацу, а `</p>` - його кінець.

- **Елементи:** Елемент - це повна конструкція, яка включає в себе відкриваючий тег, контент та закриваючий тег. Наприклад, `<h1>Заголовок</h1>` - це елемент заголовку першого рівня.

- **Атрибути:** Атрибути надають додаткові властивості елементам та визначають їхню поведінку.

- Наприклад, `Посилання` має атрибут `href`, який вказує на URL.

- **Структура документа:** HTML-документ має структуровану форму, включаючи елементи, такі як `<head>`, `<title>`, і `<body>`. Вони допомагають визначити заголовок документа, його метадані та вміст.

- **Вбудовані елементи:** HTML має вбудовані елементи для вставки медіа, форм, таблиць, списків та іншого контенту, що робить його універсальним для створення різних типів веб-сторінок.

- Також, ось деякі з найважливіших переваг HTML:

- **Простота використання:** HTML відомий своєю легкістю та простотою використання. Навіть початківці можуть швидко навчитися базовим концепціям та почати створювати веб-сторінки.

- **Крос-платформенність:** Веб-сторінки, розроблені з використанням HTML, можуть бути відображені на різних пристроях та операційних системах без будь-яких значущих змін.

- **Широкий спектр підтримки:** Майже всі сучасні браузері підтримують HTML, що робить його стандартом для веб-розробки. Це забезпечує єдність у відображенні вмісту на різних платформах.

- **Семантика та доступність:** HTML надає семантичні теги, які дозволяють розробникам структурувати вміст і надавати вмісту значення. Це покращує доступність вмісту для користувачів інвалідів та покращує індексацію пошуковими системами.

– **Легкість інтеграції:** HTML легко інтегрується з іншими технологіями веб-розробки, такими як CSS для стилізації та JavaScript для динамічних взаємодій, що дозволяє розробникам створювати повноцінні веб-додатки.

– **Підтримка мультимедіа:** HTML дозволяє вставляти зображення, аудіо та відео безпосередньо у вміст сторінки, що робить можливим створення різноманітних мультимедійних веб-сайтів.

– **Стандартизація:** Як стандарт, HTML сприяє стандартизації та сумісності між різними реалізаціями браузерів, що дозволяє однаково виглядати та працювати веб-сторінкам на різних платформах.

Отже, враховуючи всі переваги та незамінність HTML, він є обов'язковим про розробці веб-додатків чи веб-сайтів.

CSS

Ще одним невід'ємним інструментом при розробці веб-додатків є CSS. CSS є мовою стилізації, що використовується для визначення вигляду та форматування веб-сторінок, написаних мовою HTML. Він розділяє структуру вмісту веб-сторінки від її дизайну, дозволяючи розробникам створювати стильні та привабливі веб-сайти[24].

Переваги CSS:

– **Розділення вмісту та стилю:** CSS дозволяє відокремлювати опис стилю від HTML-коду, що полегшує розуміння та обслуговування коду.

– **Каскадна модель:** Каскадна модель CSS дозволяє налаштовувати стилі елементів, враховуючи їхнє спадкове співвідношення та порядок застосування.

– **Легке форматування та масштабування:** CSS надає розробникам гнучкість у форматуванні сторінок, що робить його легким для змін та масштабування.

– **Можливості адаптивного дизайну:** За допомогою CSS можна легко створювати адаптивні дизайни, які пристосовуються до різних розмірів екрану та пристроїв.

- **Велика кількість властивостей:** CSS надає широкий набір властивостей для стилізації тексту, кольорів, меж, фонів, позицій елементів та багато іншого.
- **Підтримка анімацій та переходів:** CSS дозволяє створювати анімації та плавні переходи між станами елементів, покращуючи користувацький досвід.
- **Сумісність з іншими технологіями:** CSS легко інтегрується з іншими технологіями, такими як HTML для створення веб-сторінок та JavaScript для створення динамічного вмісту.
- **Підтримка друкованих стилів:** CSS надає можливості для форматування та стилізації сторінок при друку, що робить його корисним для створення друкованих документів.

CSS Modules

Оскільки проект є об'ємним то також було використано таку методологію як CSS modules. CSS Modules є методологією для написання CSS коду в масштабованих і підтримуваних проектах. Вони дозволяють обмежити область дії стилів до конкретного компонента або модуля, уникнути конфліктів імен та забезпечити ізоляцію стилів.

Основні ідеї CSS Modules включають у себе:

- **Локальні імена:** Кожен модуль (компонент) має свій власний унікальний простір імен для стилів. Імена класів автоматично генеруються із унікальним ідентифікатором для забезпечення ізоляції.
- **Зв'язок між класами та компонентами:** Класи, що визначені в рамках модуля, доступні тільки в межах цього модуля. Забезпечує чітке визначення того, які стилі використовуються в конкретному компоненті.
- **Зручний синтаксис:** Використовується спеціальний синтаксис для імпортування імен стилів у JavaScript-файли.

Переваги CSS modules:

- Ізоляція стилів для кожного компонента.
- Можливість використовувати звичайний CSS.
- Підтримка автопрефіксів та інших CSS-функцій.

Недоліки CSS modules:

- Може вимагати налаштування збірника.
- Додаткові файли для стилів, що може збільшити обсяг проекту.

Враховуючи ці фактори, CSS Modules є потужним інструментом для підтримки модульності, уникнення конфліктів імен, і полегшення управління стилізацією великих та складних веб-проектів яким і є веб-додаток соціальної мережі.

Java Script

В якості мови програмування для розробки веб-додатку соціальної мережі була обрана мова Java Script. Розглянемо, що собою представляє ця мова програмування, та які її переваги при розробці веб-додатку соціальної мережі спонукали обрати саме її.

JavaScript (JS) - це високорівнева, інтерпретована мова програмування, яка широко використовується для розробки динамічних та інтерактивних веб-сайтів. Вона була створена для надання можливостей взаємодії з користувачем на стороні клієнта браузера[25].

Переваги JavaScript та її застосування для розробки соціальної мережі:

- **Клієнтська взаємодія:** JavaScript дозволяє створювати взаємодію з користувачем без перезавантаження сторінки, що робить його ідеальним для розробки соціальних мереж з динамічним та живим змістом.

- **Асинхронність:** Модель асинхронності JavaScript дозволяє взаємодіяти з сервером без блокування інтерфейсу користувача, що важливо для роботи з великою кількістю запитів, які можуть виникати у соціальних мережах.

- **Розширені можливості браузера:** JavaScript може взаємодіяти з DOM (Document Object Model) браузера, дозволяючи динамічно змінювати вміст та структуру сторінки.

- **Багатофункціональність:** JavaScript підтримує різноманітні фронтенд-фреймворки та бібліотеки, які полегшують розробку складних інтерфейсів та додаткових функцій для соціальних мереж.

– **Розширення можливостей за допомогою бібліотек і фреймворків:** Наявність таких бібліотек як React, Angular, або Vue.js дозволяє швидко створювати ефективні та високопродуктивні веб-додатки, в тому числі соціальні мережі.

– **Безпека:** Використання JavaScript на стороні клієнта дозволяє ефективно виконувати перевірки безпеки та контрольовано обробляти дані користувача перед їх відправкою на сервер.

– **Спрощена робота з API:** JavaScript легко взаємодіє з API, що дозволяє отримувати та надсилати дані на сервер і використовувати їх у реальному часі.

– **Робота на багатьох платформах:** JavaScript може виконуватися на багатьох платформах та браузерях, що робить його універсальним для веб-розробки.

Враховуючи всі перераховані переваги даної мови програмування при розробці веб-додатку соціальної мережі, було обрано саме її.

3.1.3 ОГЛЯД ПРОГРАМНОЇ ПЛАТФОРМИ

Node.js

В якості програмної платформи для виконання JS коду на сервері була обрана Node.js.

Node.js - це відкрите середовище виконання JavaScript на стороні сервера. Воно базується на двигуні V8 від Google та забезпечує ефективний спосіб створення масштабованих та швидких серверних додатків. Однією з ключових особливостей Node.js є асинхронний та подійно-орієнтований підхід, що дозволяє ефективно обробляти багато одночасних з'єднань[26].

Логотип Node.js представлено на рис. 3.1.2.



Рисунок 3.1.2 Логотип Node.js

Чому було обрано саме Node.js для розробки веб-додатку соціальної мережі:

– **Асинхронність та ефективність:** Модель асинхронного виконання Node.js дозволяє ефективно обробляти багато запитів одночасно, що є важливим для соціальних мереж з великою кількістю користувачів та одночасних взаємодій.

– **Використання JavaScript на обох сторонах:** Можливість використовувати JavaScript як на стороні клієнта (фронтенд), так і на стороні сервера (бекенд), спрощує вирішення завдань та обмін коду між клієнтом та сервером.

– **Широка підтримка бібліотек та фреймворків:** Існує багато бібліотек та фреймворків, таких як Express.js, які полегшують створення веб-додатків на Node.js.

– **Потужний пакетний менеджер npm:** npm (Node Package Manager) є великим репозиторієм для сторонніх бібліотек та пакетів, що полегшує використання готових рішень та їхню інтеграцію у проект.

– **Підтримка реального часу:** Node.js ідеально підходить для розробки функціоналу в реальному часі, такого як миттєві сповіщення, онлайн-чати, та інші елементи соціальної взаємодії.

– **Гнучкість та масштабованість:** Node.js дозволяє гнучко налаштувати серверну частину соціальної мережі та легко масштабувати її з ростом користувачів.

Отже, вибір Node.js для розробки соціальної мережі дозволяє створювати ефективні та швидкодіючі застосунки з акцентом на реальний час та велику кількість одночасних взаємодій.

3.1.4 ОГЛЯД ВИКОРИСТАНИХ БІБЛІОТЕК ТА ФРЕЙМВОРКІВ

React

Для розробки фронтенд частини веб-додатку в якості основної бібліотеки було обрано React.

React – це бібліотека JavaScript, створена компанією Facebook, яка використовується для розробки інтерфейсів користувача. Вона дозволяє розробникам створювати компоненти, які автоматично оновлюються при зміні стану додатку, що робить код більш ефективним та простим у управлінні.

Логотип React представлено на рис. 3.1.3.



Рисунок 3.1.3 Логотип React

React – це найбільш популярний фреймворк за 2022 рік[27].

Рейтинг топ-10 фронтенд фреймворків представлено на рис. 3.1.4.

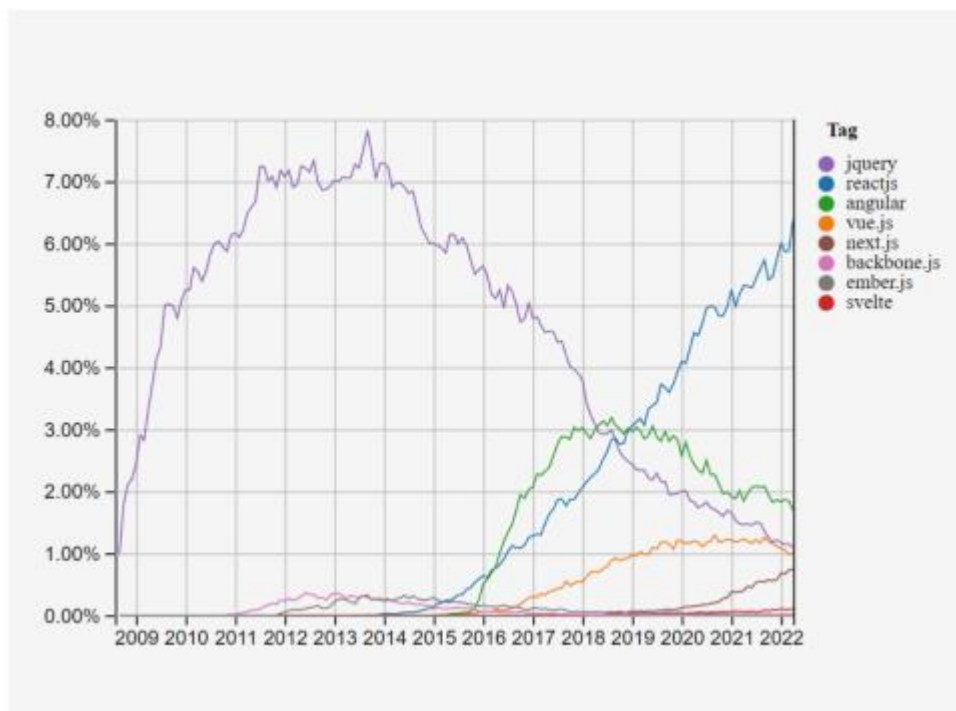


Рисунок 3.1.4 Рейтинг топ-10 фронтенд фреймворків станом на 2022 рік

Чому слід обрати React в якості бібліотеки для розробки веб-додатку соціальної мережі[28, 29]:

– **Компонентна архітектура:** React базується на компонентній архітектурі, що дозволяє створювати відокремлені компоненти, якими легко управляти та повторно використовувати. Це особливо корисно для створення складних інтерфейсів соціальної мережі.

– **Віртуальний DOM:** Використання віртуального DOM дозволяє React ефективно оновлювати лише ті елементи сторінки, які змінилися, забезпечуючи високу продуктивність та швидке відображення змін.

– **JSX - розширення JavaScript:** JSX, яке використовується в React, дозволяє писати компоненти в синтаксисі, що нагадує HTML. Це полегшує розробку та розуміння коду.

– **Односторінкові додатки:** React ідеально підходить для створення односторінкових додатків (SPA), які є динамічними та ефективними у використанні.

- **Функціональні компоненти та хуки:** Введення функціональних компонентів та хуків (наприклад, `useState`, `useEffect`) робить код React більш простим та зрозумілим.

- **Підтримка реактивного підходу:** З React легко інтегрувати реактивні бібліотеки для роботи з реальним часом та обміном даними в соціальних мережах.

Обираючи React, можна швидко створити динамічний та ефективний інтерфейс для соціальної мережі, використовуючи сучасні техніки розробки.

Redux

Для керування станами в розроблюваному веб-додатку соціальної мережі було використано бібліотеку Redux.

Redux - це бібліотека для керування станом додатка в JavaScript-додатках. Вона широко використовується в React-додатках, але може бути використана з іншими бібліотеками або фреймворками[30].

Основні концепції Redux:

- **Стор:** Стор (Store) - це об'єкт, який зберігає стан вашого додатка. Всі дані додатка знаходяться в одному сторі, що полегшує керування та відстеження стану.

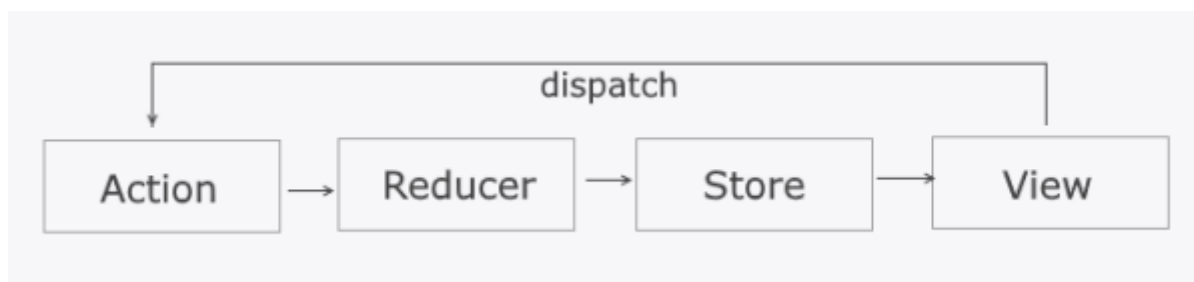
- **Дії:** Дії (Actions) - це об'єкти, які вказують, що відбувається у вашому додатку. Вони відправляються до стора і викликають зміни стану.

- **Редуктори:** Редуктори (Reducers) - це функції, які обробляють дії та змінюють стан додатка. Кожен редуктор відповідає за частину стану.

- **Сполучний Редуктор:** Сполучний редуктор (Combined Reducer) - це функція, яка об'єднує всі редуктори в один, щоб вони могли бути використані стором.

- **Середовище Дій:** Середовище дій (Action Creators) - це функції, які створюють дії. Це робить код чистішим та дозволяє змінювати логіку створення дій без прямого впливу на самі дії.

Загальну схему[31] взаємодії елементів архітектури Redux представлено на рис. 3.1.5.



3.1.5 Загальна схема взаємодії елементів архітектури Redux

Отже, Використання Redux у веб-додатках соціальної мережі дозволяє покращити організацію та керування станом, що важливо в умовах складної взаємодії та обміну даними між різними компонентами.

Bootstrap

Bootstrap - це відкритий набір інструментів для розробки веб-додатків інтерфейсів. Він містить HTML, CSS та JavaScript-компоненти, які спрощують розробку та роблять веб-сайти більш адаптивними та стильними. Bootstrap був створений командою Twitter і швидко став популярним серед розробників завдяки своїй простоті та ефективності. Логотип Bootstrap представлено на рис. 3.1.6.



3.1.6 Логотип Bootstrap

При виборі Bootstrap для визначення який саме інструмент краще використати, було проведено порівняння з іншим популярним рішенням таким як Tailwind CSS. Популярність порівнювальних фреймворків за останні 5 років зображено на рис. 3.1.7

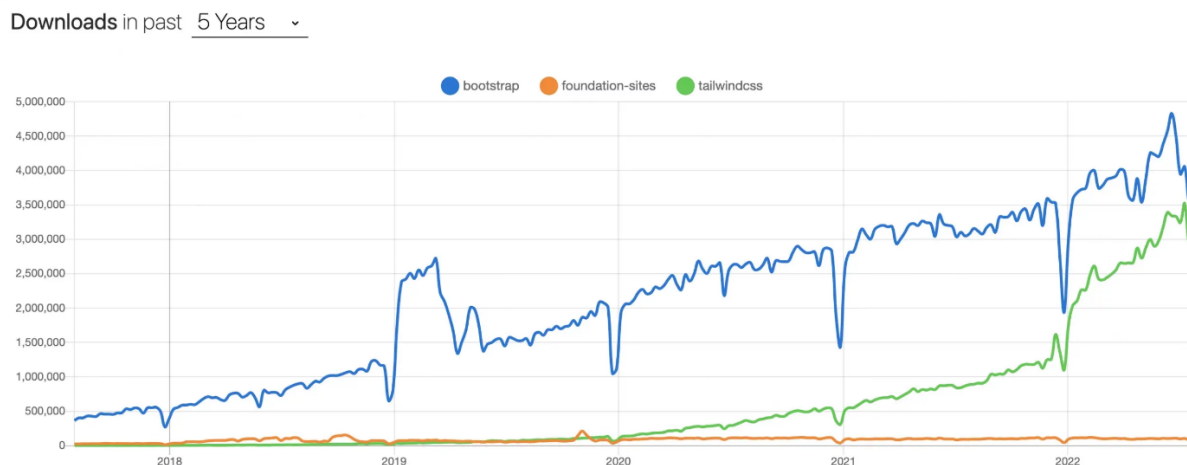


Рисунок. 3.1.7. Порівняльний графік фреймворків

Порівняння Bootstrap та Tailwind CSS за характеристиками:

Дизайн і Стилі

- Bootstrap: Має власний дизайн та стилі. Це може бути перевагою, якщо розробникам подобається консистентний вигляд та швидке впровадження.
- Tailwind CSS: Не має власного дизайну. Потрібно буде самостійно створювати стилі. Це дозволить максимально контролювати вигляд вашого проекту.

Вивчення та Використання

- Bootstrap: Легко вивчається, заздалегідь знаючи класи, можна швидко впроваджувати стилі та компоненти.
- Tailwind CSS: Вимагає часу на вивчення та адаптацію. Використання "utility-first" підходу може здатися нетрадиційним для тих, хто звик до інших методів.

Налаштування

- Bootstrap: Має обмежені можливості налаштування. Готовий стиль може виглядати однаково для багатьох проектів.

– Tailwind CSS: Надає гнучкість та повний контроль над стилями. Ви визначаєте, які саме стилі використовувати.

JavaScript

– Bootstrap: Містить JavaScript-компоненти для модальних вікон, каруселей та інших інтерактивних елементів.

– Tailwind CSS: Фокусується лише на CSS і не включає в себе готових JavaScript-рішень.

Розмір

– Bootstrap: Зазвичай має більший обсяг файлів, оскільки включає готовий дизайн і компоненти.

– Tailwind CSS: Може мати менший розмір файлів, оскільки ви використовуєте лише ті класи, які потрібні для вашого проекту.

Отже, основними перевагами Bootstrap можна виділити його швидке впровадження та менший час зайнятий на розробку, що ідеально підходить для розроблюваного веб-додатку соціальної мережі.

Bootstrap в даному проекті був використаний для зменшення використання власного коду, використання готових компонентів, та для використання адаптивного дизайну.

Express.js

Для розробки серверної частини веб-додатку був використаний фреймворк Express.js.

Express.js - це веб-фреймворк для Node.js, який полегшує створення веб-додатків та API. Express забезпечує спрощений спосіб обробки HTTP-запитів та взаємодії з базами даних. Він є одним з найпопулярніших фреймворків для створення серверної частини додатків на платформі Node.js.

Основні компоненти та можливості Express.js:

– **Маршрутизація (Routing):** Express дозволяє легко визначати маршрути для обробки різних типів HTTP-запитів (GET, POST, PUT, DELETE тощо). Маршрутизація допомагає визначити, які дії виконувати для конкретних URL.

– **Мідлвари (Middleware):** Мідлвари в Express - це функції, які мають доступ до об'єкта запиту (req), об'єкта відповіді (res), та наступної функції у циклі запит-відповідь (зазвичай позначеної як next). Мідлвари використовуються для виконання певних операцій перед обробкою маршрутів або після неї.

– **Шаблонізатори (Template Engines):** Express може інтегрувати різні шаблонізатори (наприклад, EJS, Pug) для генерації HTML-сторінок на сервері та їх відправлення на клієнт.

– **Статичні файли (Static Files):** Express дозволяє обслуговувати статичні файли (зображення, стилі, скрипти) заздалегідь визначеними каталогами, щоб полегшити доступ до них з браузера.

– **Обробка POST-запитів та дані форм:** Express дозволяє легко обробляти POST-запити та отримувати дані форм. Це важливо для обробки даних, які користувачі вводять у веб-форми.

Використання Express.js для розробки веб-додатку соціальної мережі:

– **Створення API:** Express використовують для створення API, яке використовується фронтендом соціальної мережі для обміну даними з сервером.

– **Маршрутизація Користувачів:** Express може бути використаний для визначення маршрутів, пов'язаних з користувачами (наприклад, реєстрація, вхід, профіль).

– **Робота з Базою Даних:** За допомогою Express можна легко здійснювати взаємодію з базами даних, де зберігаються дані про користувачів та їх взаємодії.

– **Аутифікація та Авторизація:** Express може використовуватися для реалізації систем аутифікації та авторизації, які є важливими в соціальних мережах для захисту інформації користувачів.

Express.js дозволяє швидко створювати потужні серверні частини для веб-додатків соціальної мережі і забезпечує ефективну обробку HTTP-запитів та взаємодію з різними компонентами додатку.

Json Web Token

JWT (JSON Web Token) - це стандарт, який визначає компактний та самостійний спосіб безпечного передавання інформації між сторонами як об'єкт JSON. Ця інформація може бути перевірена та довірена, оскільки JWT підписується (signs) за допомогою секретного ключа або публічного/приватного ключа (за допомогою алгоритму підпису)[32].

JWT складається з трьох частин:

1) **Header (Заголовок):** Містить тип токена та тип підпису (наприклад, HMAC SHA256 або RSA).

2) **Payload (Навантаження):** Містить дані (корисну інформацію), які ви хочете передати. Наприклад, ім'я користувача чи роль.

3) **Signature (Підпис):** Підпис, який генерується на основі заголовка та навантаження, використовуючи секретний ключ. Це дозволяє перевірити, чи дані не були змінені під час передачі.

JWT використовується в розробці веб-додатків соціальних мереж, зокрема для наступних випадків:

- **Аутентифікація:** Після успішної аутентифікації користувача сервер може видати JWT, який містить інформацію про користувача (наприклад, його ідентифікатор, роль тощо). Цей токен може бути включений у кожен подальший запит, і сервер може перевірити його валідність та визначити, які ресурси або функції доступні цьому користувачеві.

- **Авторизація:** JWT може також містити інформацію про те, на які ресурси або функції користувач має доступ. Це дозволяє забезпечити контроль доступу та авторизацію на рівні токена.

- **Захист інформації:** Так як JWT підписується, він дозволяє забезпечити захист від зміни даних під час передачі. Це важливо для забезпечення цілісності інформації про користувача та його дозволів.

Bcrypt

bcrypt - це алгоритм хешування паролів, спеціально розроблений для зберігання паролів в безпечному вигляді. Однією з його головних особливостей є унікальний спосіб зберігання паролів, який робить важко проводити атаки, спрямовані на отримання оригінального паролю навіть при витоку хеш-значень[33].

Основні характеристики bcrypt:

– **Характеристика Витрати Ресурсів (Resource Cost):** Алгоритм bcrypt залежить від великої кількості ресурсів (CPU та пам'яті), що робить його більш витратним у порівнянні з простішими алгоритмами хешування. Це ускладнює атаки на основі перебору.

– **Сіль (Salt):** Кожен пароль хешується з унікальним і випадковим значенням, відомим як "сіль" (salt). Використання солі ускладнює атаки різних видів, таких як атаки за допомогою таблиць райдужниць (rainbow tables).

– **Адаптивність (Adaptability):** Bcrypt має вбудовану адаптивність, яка дозволяє змінювати кількість ітерацій в залежності від доступності ресурсів. Це означає, що при виникненні обчислювальних ресурсів, необхідних для власності, можна збільшити кількість ітерацій для ускладнення атак.

– **Усталеність (Staleness):** Bcrypt вбудовано в опір старінню, що означає, що його безпека не погіршується з часом, оскільки можна змінювати кількість ітерацій.

У веб-додатках соціальних мереж, де безпека паролів користувачів є важливою, використання bcrypt для хешування та зберігання паролів є стандартним практичним підходом. При авторизації користувача пароль вводиться, хешується за допомогою bcrypt, і порівнюється з збереженим хешем у базі даних.

Socket.IO

Для реалізації подій в реальному часі які є дуже важливими в соціальних мережах, було використано бібліотеку Socket.IO.

Socket.IO - це бібліотека для роботи з реальним часом у веб-додатках. Вона побудована на основі WebSocket, але автоматично переходить на інші транспортні протоколи, такі як HTTP long polling, при необхідності. Socket.IO дозволяє відповідати на події в реальному часі, відсилаючи та отримуючи події між клієнтом та сервером[34].

Основні характеристики Socket.IO:

– **Реальний Час:** Socket.IO дозволяє надсилати та приймати дані в режимі реального часу, що корисно для миттєвого оновлення вмісту на сторінці без необхідності перезавантаження.

– **Події:** Використовуючи Socket.IO, ви можете визначати та обробляти події (events), які виникають як на клієнті, так і на сервері.

– **Підтримка Різних Транспортів:** Використовуючи різні транспортні протоколи (WebSocket, HTTP long polling тощо), Socket.IO забезпечує максимальну сумісність з різними браузерами та умовами мережі.

– **Простота Використання:** Socket.IO надає простий API для створення та обробки з'єднань, відсилання та прийому подій.

– **Автоматичне Відновлення З'єднань:** Socket.IO автоматично спробує відновити з'єднання при втраті зв'язку або мережі.

– При розробці веб-додатку соціальної мережі Socket.IO використано для таких сценаріїв:

– **Миттєві Оновлення:** Сповіщення користувачів про нові повідомлення, коментарі, лайки та інші події в реальному часі.

– **Чат та дзвінки:** Створення системи чату для приватних обговорень та система аудіо та відео дзвінків.

– **Оновлення Статусу:** Миттєве оновлення статусу користувача або події в його активності.

– **Сповіщення:** Відправлення сповіщень та повідомлень безпосередньо на клієнтські пристрої.

– **Синхронізація в Реальному Часі:** Забезпечення реального часу для синхронізації вмісту між різними користувачами.

Socket.IO дозволяє створювати динамічні та взаємодіючі функції у веб-додатках соціальних мереж, забезпечуючи високопродуктивну та ефективну роботу в режимі реального часу.

3.1.5 ОГЛЯД ОБРАНОЇ БАЗИ ДАНИХ

MongoDB

MongoDB - це документо-орієнтована база даних, яка використовує формат документа JSON-подібний, відомий як BSON (Binary JSON). MongoDB є однією з нереляційних (NoSQL) баз даних і призначена для зберігання та обробки великих обсягів неструктурованих даних[35].

Логотип MongoDB представлено на рис. 3.1.8.



3.1.8 Логотип MongoDB

Основні особливості через які була обрана сама ця база даних:

– **Гнучкість Схеми:** У соціальних мережах багато різних видів даних, і гнучкість схеми MongoDB дозволяє легко додавати або змінювати дані в залежності від розвитку функціоналу.

– **Динамічні Запити:** Соціальні мережі зазвичай вимагають різних видів запитів, і MongoDB дозволяє легко створювати динамічні та складні запити до бази даних.

– **Масштабованість:** MongoDB може бути легко масштабована, що важливо для соціальних мереж з великою кількістю користувачів та даних.

– **Індексація та Швидкість:** MongoDB надає можливості для оптимізації запитів та швидкого доступу до даних, що є ключовим аспектом для додатків з багатою функціональністю соціальної мережі.

– **Зручна Мова Запитів:** Використання JavaScript-подібної мови запитів робить роботу з даними більш зручною та зрозумілою для розробників.

MongoDB є чудовим вибором бази даних для розробки веб-додатку соціальної мережі, враховуючи вимоги до даних, типи операцій та потреби у масштабованості.

3.2 ОПИС МОДУЛІВ СЕРВЕРНОЇ ЧАСТИНИ ВЕБ-ДОДАТКУ СОЦІАЛЬНОЇ МЕРЕЖІ

3.2.1 SERVER.JS

Цей модуль розгортає сервер для веб-додатка соціальної мережі, який використовує Express для обробки запитів, Mongoose для взаємодії з базою даних MongoDB, Socket.IO та Peer.js для роботи з реальним часом та обробки WebSocket-з'єднань, а також різні маршрутизатори для API ендпойнтів. Розглянемо частину цього модулю.

Цей код налаштовує маршрутизацію для різних частин API. Розглянемо цю частину модулю на рис. 3.2.1.

```
app.use('/api', require('./routes/authRouter'))
app.use('/api', require('./routes/userRouter'))
app.use('/api', require('./routes/postRouter'))
app.use('/api', require('./routes/commentRouter'))
app.use('/api', require('./routes/notifyRouter'))
app.use('/api', require('./routes/messageRouter'))
```

Рисунок 3.2.1 Маршрутизація різних частин API

Кожна стрічка `app.use('/api', require('./routes/...'))` вказує на те, що для всіх маршрутів, які починаються з `/api`, використовуватиметься відповідний роутер, який імпортується з вказаних файлів (`authRouter`, `userRouter`, і так далі).

– `app.use('/api', require('./routes/authRouter'))`: Реєструє роутер для автентифікації (логін, реєстрація та інші пов'язані операції).

– `app.use('/api', require('./routes/userRouter'))`: Реєструє роутер, пов'язаний з операціями користувача (отримання інформації про користувача, оновлення профілю і т.д.).

– `app.use('/api', require('./routes/postRouter'))`: Реєструє роутер для операцій з постами (створення, читання, оновлення, видалення).

– `app.use('/api', require('./routes/commentRouter'))`: Реєструє роутер для операцій з коментарями до постів.

– `app.use('/api', require('./routes/notifyRouter'))`: Реєструє роутер для управління сповіщеннями.

– **app.use('/api', require('./routes/messageRouter'))**: Реєструє роутер для обробки повідомлень між користувачами.

3.2.2 AUTHROUTER.JS

Цей модуль визначає роутер для операції автентифікації. Давайте розглянемо його детальніше на рис. 3.2.2.

```
const router = require('express').Router()
const authCtrl = require('../controllers/authCtrl')

router.post('/register', authCtrl.register)

router.post('/login', authCtrl.login)

router.post('/logout', authCtrl.logout)

router.post('/refresh_token', authCtrl.generateAccessToken)

module.exports = router
```

Рисунок 3.2.2 Код модулю authRouter.js

– **const router = require('express').Router()**: Створюється екземпляр роутера Express.

– **const authCtrl = require('../controllers/authCtrl')**: Підключається модуль authCtrl, який містить контролери для операцій реєстрації, входу, виходу та оновлення токена.

– **router.post('/register', authCtrl.register)**: Визначає маршрут для обробки POST-запиту на /register та вказує на відповідний контролер authCtrl.register. Цей маршрут призначений для реєстрації нового користувача.

– **router.post('/login', authCtrl.login)**: Визначає маршрут для обробки POST-запиту на /login та вказує на відповідний контролер authCtrl.login. Цей маршрут призначений для входу користувача.

– **router.post('/logout', authCtrl.logout)**: Визначає маршрут для обробки POST-запиту на /logout та вказує на відповідний контролер authCtrl.logout. Цей маршрут призначений для виходу користувача.

– **router.post('/refresh_token', authCtrl.generateAccessToken):** Визначає маршрут для обробки POST-запиту на /refresh_token та вказує на відповідний контролер authCtrl.generateAccessToken. Цей маршрут призначений для оновлення токена доступу.

– **module.exports = router:** Експортується створений роутер для використання його в інших частинах додатка.

Цей роутер служить для обробки різних видів аутентифікаційних запитів у веб-додатку, таких як реєстрація, вхід, вихід та оновлення токенів доступу. Контролери, які викликаються для кожного маршруту, містять бізнес-логіку для відповідних операцій автентифікації.

3.2.3 AUTHCTRL.JS

Цей модуль містить контролери для реєстрації, входу, виходу та оновлення токенів доступу в системі автентифікації. Давайте розглянемо кожен контролер окремо:

register: Цей контролер обробляє запит на реєстрацію нового користувача. Він перевіряє унікальність імені користувача та адреси електронної пошти, валідність пароля, і створює новий об'єкт користувача в базі даних. Після успішної реєстрації видається токен доступу та токен оновлення, які використовуються для автентифікації. Контролер реєстрації зображено на рис. 3.2.3.

```

register: async (req, res) => {
  try {
    const { fullname, username, email, password, gender } = req.body
    let newUserName = username.toLowerCase().replace(/ /g, '')

    const user_name = await Users.findOne({username: newUserName})
    if(user_name) return res.status(400).json({msg: "This user name already exists."})

    const user_email = await Users.findOne({email})
    if(user_email) return res.status(400).json({msg: "This email already exists."})

    if(password.length < 6)
      return res.status(400).json({msg: "Password must be at least 6 characters."})

    const passwordHash = await bcrypt.hash(password, 12)

    const newUser = new Users({
      fullname, username: newUserName, email, password: passwordHash, gender
    })

    const access_token = createAccessToken({id: newUser._id})
    const refresh_token = createRefreshToken({id: newUser._id})

    res.cookie('refreshtoken', refresh_token, {
      httpOnly: true,
      path: '/api/refresh_token',
      maxAge: 30*24*60*60*1000 // 30days
    })

    await newUser.save()

    res.json({
      msg: 'Register Success!',
      access_token,
      user: {
        ...newUser._doc,
        password: ''
      }
    })
  } catch (err) {
    return res.status(500).json({msg: err.message})
  }
}

```

Рисунок 3.2.3 Код контролеру реєстрації

login: Цей контролер обробляє запит на вхід користувача. Він перевіряє існування користувача з вказаною адресою електронної пошти, порівнює введений пароль з хешем пароля в базі даних і, у разі успіху, видає токени доступу та оновлення. Контролер логіну зображено на рис. 3.2.4.

```

login: async (req, res) => {
  try {
    const { email, password } = req.body

    const user = await Users.findOne({email})
      .populate("followers following", "avatar username fullname followers following")

    if(!user) return res.status(400).json({msg: "This email does not exist."})

    const isMatch = await bcrypt.compare(password, user.password)
    if(!isMatch) return res.status(400).json({msg: "Password is incorrect."})

    const access_token = createAccessToken({id: user._id})
    const refresh_token = createRefreshToken({id: user._id})

    res.cookie('refreshtoken', refresh_token, {
      httpOnly: true,
      path: '/api/refresh_token',
      maxAge: 30*24*60*60*1000 // 30days
    })

    res.json({
      msg: 'Login Success!',
      access_token,
      user: {
        ...user._doc,
        password: ''
      }
    })
  } catch (err) {
    return res.status(500).json({msg: err.message})
  }
}

```

Рисунок 3.2.4 Код контролеру логіну

logout: Цей контролер видаляє токен оновлення з куки, що еквівалентно виходу користувача. Контролер логату зображено на рис. 3.2.5.

```

logout: async (req, res) => {
  try {
    res.clearCookie('refreshtoken', {path: '/api/refresh_token'})
    return res.json({msg: "Logged out!"})
  } catch (err) {
    return res.status(500).json({msg: err.message})
  }
},

```

Рисунок 3.2.5 Код контролеру логату

generateAccessToken: Цей контролер викликається для оновлення токена доступу. Він перевіряє наявність токена оновлення, розшифровує його, знаходить користувача в базі даних, і видає новий токен доступу разом з

інформацією про користувача. Контролер оновлення токена доступу зображено на рис. 3.2.6.

```
generateAccessToken: async (req, res) => {
  try {
    const rf_token = req.cookies.refreshToken
    if(!rf_token) return res.status(400).json({msg: "Please login now."})

    jwt.verify(rf_token, process.env.REFRESH_TOKEN_SECRET, async(err, result) => {
      if(err) return res.status(400).json({msg: "Please login now."})

      const user = await Users.findById(result.id).select("-password")
        .populate('followers following', 'avatar username fullname followers following')

      if(!user) return res.status(400).json({msg: "This does not exist."})

      const access_token = createAccessToken({id: result.id})

      res.json({
        access_token,
        user
      })
    })
  } catch (err) {
    return res.status(500).json({msg: err.message})
  }
}
```

Рисунок 3.2.6 Код контролеру оновлення токена

Весь цей набір контролерів забезпечує безпеку аутентифікації та авторизації в веб-додатку. Вони використовують бібліотеку bcrypt для хешування паролів та jsonwebtoken для створення та верифікації токенів.

3.2.4 AUTH.JS

Цей middleware, позначений як auth, призначений для перевірки автентифікації користувача за наявності токена доступу в заголовку запиту. Давайте розглянемо його детальніше на рис 3.2.7.

```

const Users = require("../models/userModel")
const jwt = require('jsonwebtoken')

const auth = async (req, res, next) => {
  try {
    const token = req.header("Authorization")

    if(!token) return res.status(400).json({msg: "Invalid Authentication."})

    const decoded = jwt.verify(token, process.env.ACCESS_TOKEN_SECRET)
    if(!decoded) return res.status(400).json({msg: "Invalid Authentication."})

    const user = await Users.findOne({_id: decoded.id})

    req.user = user
    next()
  } catch (err) {
    return res.status(500).json({msg: err.message})
  }
}

module.exports = auth

```

Рисунок 3.2.7 Код модуля auth.js

- **const Users = require("../models/userModel")**: Підключається модель користувача, щоб мати можливість виконувати операції з базою даних, такі як пошук користувача за його ідентифікатором.

- **const jwt = require('jsonwebtoken')**: Підключається бібліотека jsonwebtoken для створення та верифікації токенів.

- **const auth = async (req, res, next) => { ... }**: Визначається сам middleware. Приймає три параметри: req (запит), res (відповідь) та next (функція, яка викликається для продовження виконання наступного middleware чи маршруту).

- **const token = req.header("Authorization")**: Отримує токен доступу з заголовка "Authorization" запиту.

- **if(!token) return res.status(400).json({msg: "Invalid Authentication."})**: Перевіряє наявність токена. Якщо токен відсутній, відправляє відповідь з помилкою та статусом 400.

- **Const decoded = jwt.verify(token, process.env.ACCESS_TOKEN_SECRET):** Верифікує токен, використовуючи секретний ключ з оточення та отримує розшифровані дані з токenu.

- **if(!decoded) return res.status(400).json({msg: "Invalid Authentication."}):** Перевіряє, чи вдалося розшифрувати токен. Якщо невдало, відправляє відповідь з помилкою та статусом 400.

- **const user = await Users.findOne({_id: decoded.id}):** За допомогою розшифрованого ідентифікатора користувача шукає в базі даних відповідного користувача.

- **req.user = user:** Додає знайденого користувача до об'єкта req, щоб його можна було використовувати у наступних маршрутах.

- **next():** Викликає функцію next(), щоб продовжити виконання наступного маршруту.

Демо визначення поняттю middleware.

Middleware - це функції, які мають доступ до об'єкту запиту (req), об'єкту відповіді (res), та до функції next у циклі запит-відповідь Express. Вони використовуються для виконання деяких операцій на етапі обробки запиту перед тим, як він дійде до фактичного маршруту або контролера.

Цей middleware відіграє важливу роль у перевірці та обробці токенів доступу, що забезпечує безпеку маршрутів, які вимагають автентифікації користувача.

3.2.5 USERMODEL.JS

Цей модуль це схема користувача (userSchema) для бази даних MongoDB, яка використовує бібліотеку Mongoose для моделювання та взаємодії з базою даних. Розглянемо детальніше на рис. 3.2.8.

```

const mongoose = require('mongoose')

const userSchema = new mongoose.Schema({
  fullname: {
    type: String,
    required: true,
    trim: true,
    maxlength: 25
  },
  username: {
    type: String,
    required: true,
    trim: true,
    maxlength: 25,
    unique: true
  },
  email: {
    type: String,
    required: true,
    trim: true,
    unique: true
  },
  password: {
    type: String,
    required: true
  },
  avatar: {
    type: String,
    default: 'https://res.cloudinary.com/smirnoff/image/upload/v1696235171/xfny9nrfofmyt9osocw.png'
  },
  role: {type: String, default: 'user'},
  gender: {type: String, default: 'male'},
  mobile: {type: String, default: ''},
  address: {type: String, default: ''},
  story: {
    type: String,
    default: '',
    maxlength: 200
  },
  website: {type: String, default: ''},
  followers: [{type: mongoose.Types.ObjectId, ref: 'user'}],
  following: [{type: mongoose.Types.ObjectId, ref: 'user'}],
  saved: [{type: mongoose.Types.ObjectId, ref: 'user'}]
}, {
  timestamps: true
})

```

Рисунок 3.2.8 Код модуля UserModel.js

Давайте розглянемо кожне поле цієї схеми:

- **fullname:** Реєструє повне ім'я користувача. Має обов'язкове поле, яке не може бути порожнім, обрізається зайві пробіли та має максимальну довжину 25 символів.

- **username:** Реєструє ім'я користувача. Має обов'язкове поле, яке не може бути порожнім, обрізається зайві пробіли, має максимальну довжину 25 символів та є унікальним.

- **email:** Реєструє адресу електронної пошти користувача. Має обов'язкове поле, яке не може бути порожнім, обрізається зайві пробіли та є унікальним.
- **password:** Реєструє хеш пароля користувача. Має обов'язкове поле.
- **avatar:** Зберігає URL зображення аватара користувача. Має значення за замовчуванням, якщо користувач не вказав свій аватар.
- **role:** Визначає роль користувача (за замовчуванням 'user').
- **gender:** Визначає стать користувача (за замовчуванням 'male').
- **mobile, address, story, website:** Додаткові поля для зберігання мобільного номеру, адреси, історії та веб-сайту користувача.
- **followers, following, saved:** Зберігає масив ідентифікаторів користувачів, які підписані (followers), які користувач підписаний (following) та збережені користувачем (saved).
- **timestamps:** Включає автоматичне створення полів createdAt та updatedAt, які вказують на час створення та останнього оновлення запису в базі даних.

Ця схема визначає структуру даних для користувачів і використовується для створення об'єктів користувачів та взаємодії з базою даних MongoDB за допомогою Mongoose.

ВИСНОВОК ДО РОЗДІЛУ 3

У третьому розділі АВР було ретельно розглянуто та обґрунтовано вибір засобів розробки для створення веб-додатку соціальної мережі. При виборі середовища розробки враховувалися параметри ефективності, зручності та розширюваності. Обране середовище надає широкий спектр інструментів для розробки, тестування та налагодження веб-додатку, забезпечуючи при цьому комфортні умови для роботи розробника.

Мова програмування також була обрана з урахуванням її придатності для вирішення конкретних завдань, що виникають при створенні соціальної мережі. Обрана мова дозволяє ефективно взаємодіяти з веб-технологіями, забезпечуючи при цьому високий рівень продуктивності та гнучкість у розробці.

У виборі програмної платформи, бібліотек та фреймворків враховувалися основні вимоги до функціональності соціальної мережі. Обрані засоби надають надійні інструменти для створення високопродуктивних та масштабованих веб-додатків.

База даних була вибрана з урахуванням потреб системи у зберіганні та оптимізованому доступі до інформації. Обрана база даних відповідає вимогам до швидкодії та надійності, що є важливими факторами для успішної роботи соціальної мережі.

У розділі також були детально описані модулі серверної частини веб-додатку. Кожен модуль виконує конкретні функції, необхідні для нормального функціонування соціальної мережі.

4 ТЕСТУВАННЯ ІНТЕРФЕЙСУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В даній АВР було розроблено веб-додаток соціальної мережі. При розробці було використано мову програмування JavaScript, середовище розробки Visual Studio Code, програмну платформу Node.js, для розробки інтерфейсу користувача було використано бібліотеку React, фреймворк Bootstrap, CSS та HTML, бібліотеку Redux для управління станами, для розробки серверної частини додатку було використано фреймворк Express.js, для взаємодії з WebSocket та реалізації чату, дзвінків було використано бібліотеку Socket.io, для забезпечення безпеки користувача були використані такі бібліотека як Bcrypt, для створення токенів доступу JsonWebToken.

В якості тестового хостингу ми використовуємо локальний хостинг з адресом localhost та портом :3000 для фронтенд частини, та localhost з портом :5000 для бекенд частини. В майбутньому на фазі впровадження сайту в простір Інтернету можна буде використати безкоштовний хостинг Heroku або придбати хостинг деінде.

Результати розробки та інтерфейс користувача зображено на рис. 4.1-4.27

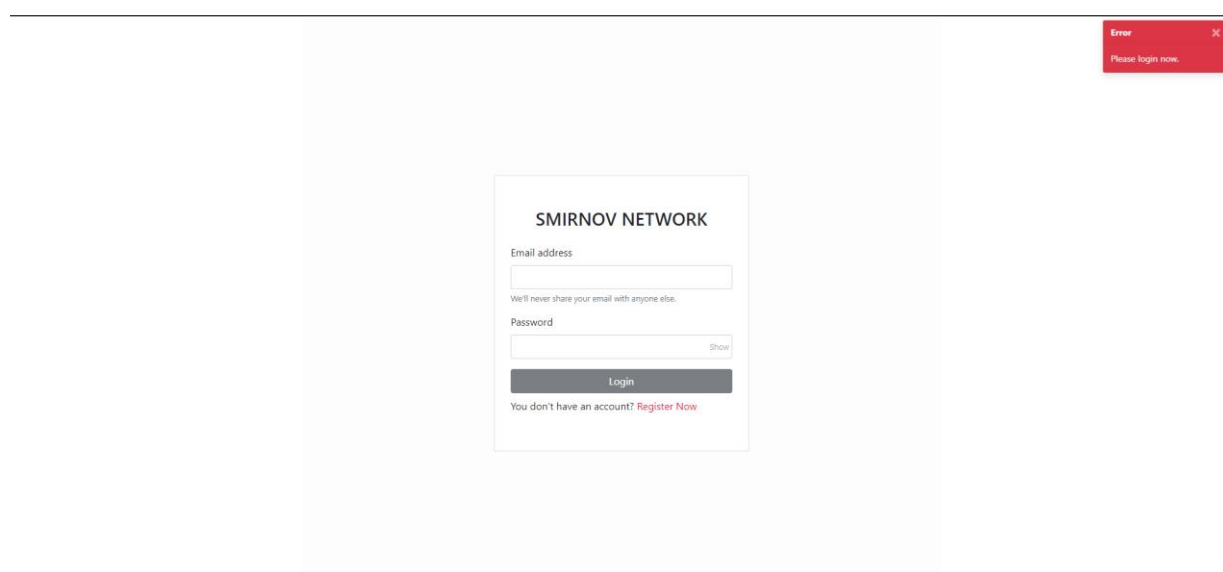
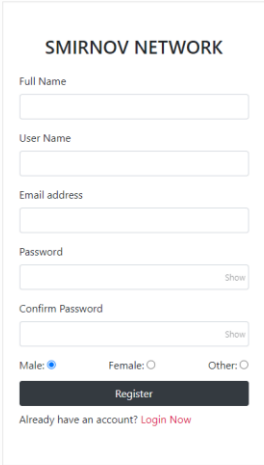


Рисунок 4.1 Сторінка авторизації

Для користування веб-додатком користувач обов'язково повинен пройти авторизацію, на цьому рисунку зображено сторінку авторизації на якій користувач може ввійти в вже існуючий акаунт, або зареєструвати новий.



The image shows a registration form titled "SMIRNOV NETWORK". The form contains the following fields and elements:

- Full Name:
- User Name:
- Email address:
- Password: with a "Show" link.
- Confirm Password: with a "Show" link.
- Gender selection: Male: , Female: , Other:
- Register button: A dark button with the text "Register".
- Link: "Already have an account? [Login Now](#)"

Рисунок 4.2 Сторінка реєстрації

При переході на сторінку реєстрації користувачу буде відображена форма в яку він повинен ввести свої дані, які є обов'язковими для реєстрації. А саме своє прізвище та ім'я, нік, пошту, пароль, підтвердити пароль та вибрати свою стать.

Також якщо користувачем будуть введені некоректні дані, або вже такий користувач існує, спрацює валідація та повідомить користувачу в чому саме помилка при реєстрації.

SMIRNOV NETWORK

Full Name

Please add your full name.

User Name

Please add your user name.

Email address

Please add your email.

Password
 Show

Confirm Password
 Show
Confirm password did not match.

Male: Female: Other:

Already have an account? [Login Now](#)

Рисунок 4.3 Валідація користувач увів не всі дані

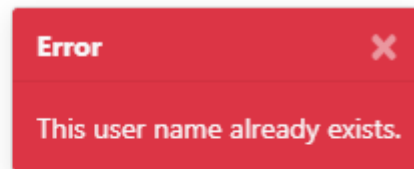


Рисунок 4.4 Валідація користувач з таким ніком вже існує

The image shows a registration form titled "SMIRNOV NETWORK". The form contains the following fields and elements:

- Full Name:** Input field with the value "Владислав Смірнов".
- User Name:** Input field with the value "Smirnov".
- Email address:** Input field with the value "12347".
- Confirm Password:** Input field with the value "Kaka133712".
- Gender:** Radio buttons for "Male" (selected), "Female", and "Other".
- Buttons:** A "Register" button and a "Login Now" link.
- Validation Error:** A red message box above the email field states: "Адрес электронной почты должен содержать символ '@'. В адресе '12347' отсутствует символ '@'." (The email address must contain the symbol '@'. The address '12347' does not contain the symbol '@').

Рисунок 4.5 Валідація користувач ввів некоректну пошту

Після успішної реєстрації користувач автоматично перейде на головну сторінку, а в правому верхньому кутку буде відображено повідомлення що користувач успішно зареєструвався.

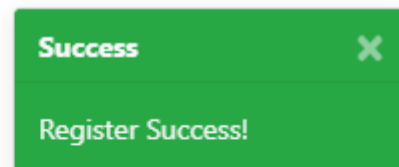


Рисунок 4.6 Успішна реєстрація

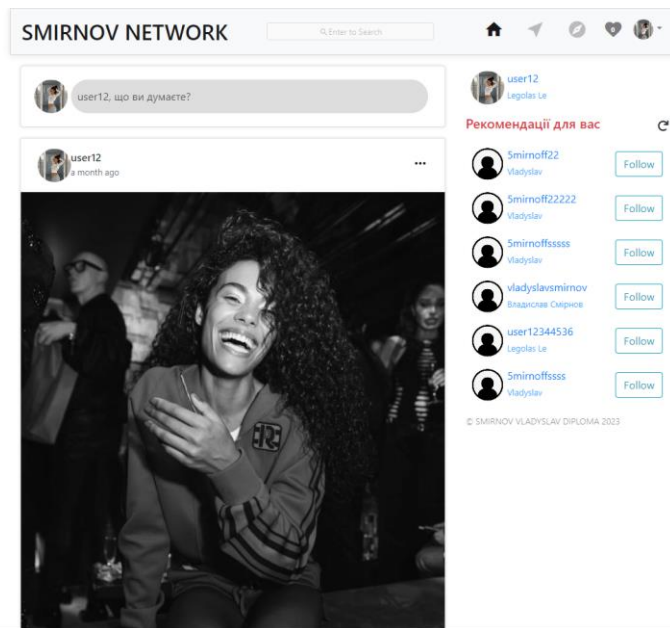


Рисунок 4.7 Головна сторінка веб-додатку

На головній сторінці користувачу будуть відображені його дописи, або дописи людей на яких він підписаний, також форма для створення нового допису, рекомендації, NavBar на якому є пошук, повідомлення, сповіщення, профіль користувача та сторінка з цікавим.

При натисканні в NavBar на профіль користувача буде відображено список доступних дій для користувача, а саме перейти до профіль користувача, змінити тему на темну, а також вийти з акаунту.

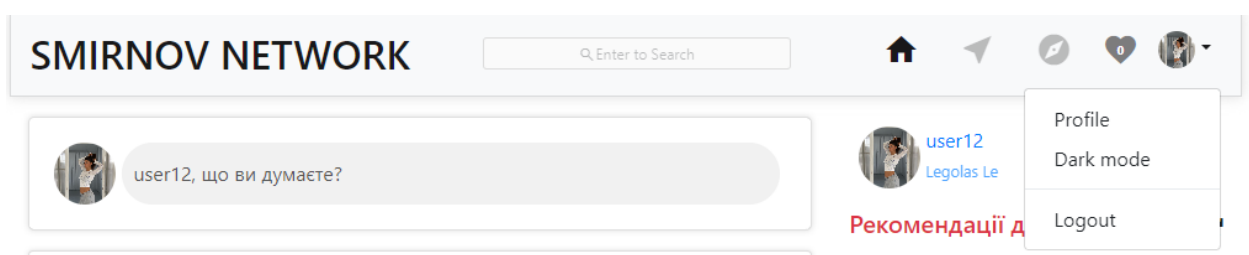


Рисунок 4.8 NavBar профіль користувача

При натисканні на темну тему, веб-додаток змінить свою тему на чорну.

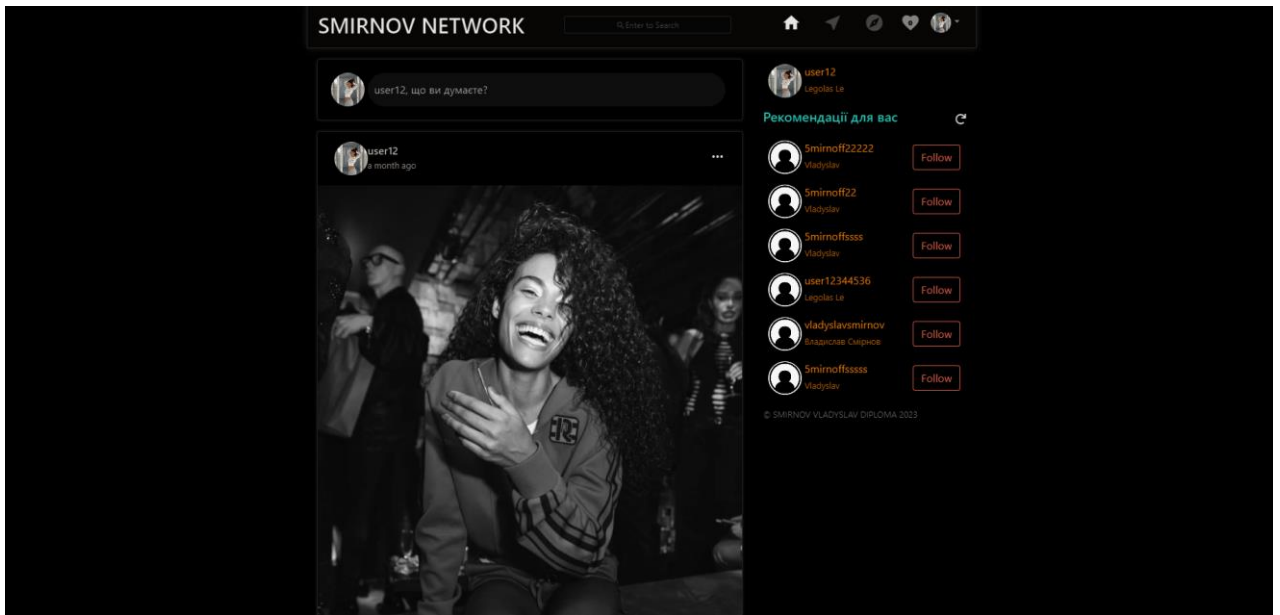


Рисунок 4.9 Головна сторінка веб-додатку при ввімкненій темній темі

При натисканні на кнопку «Профіль» користувача буде перенесено до його власного профілю, де будуть відображені всі його дописи, а також буде доступне редагування акаунту.

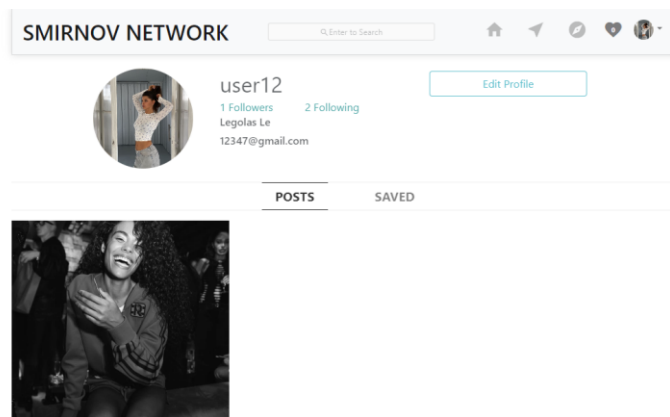


Рисунок 4.10 Сторінка профілю користувача

При натисканні на кнопку «Редагувати профіль» користувачу відкриється форма в якій він зможе змінити головне фото профілю, ім'я, додати свій

телефон, посилання на інші соціальні мережі або ресурси, місто проживання, та коротку інформацію про себе.

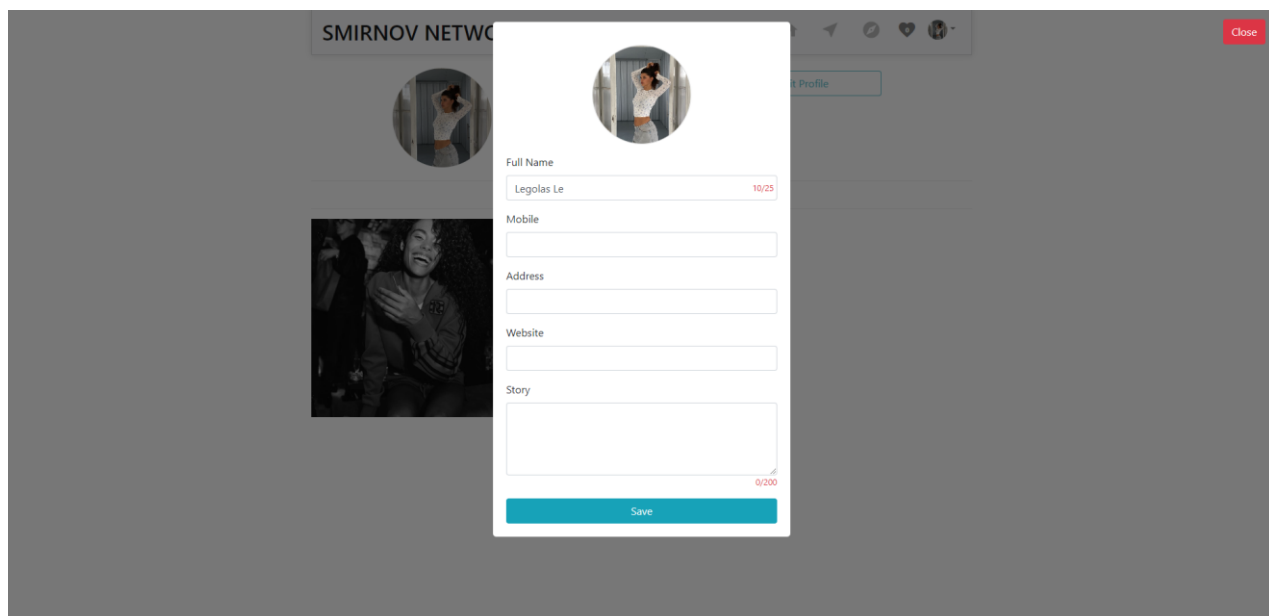


Рисунок 4.11 Редагування профілю користувача

Після додавання додаткової інформації вона буде відображена під ніком користувача.

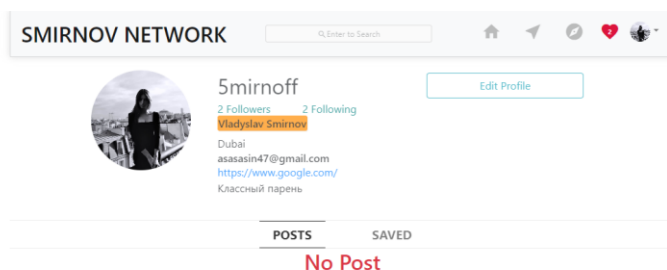


Рисунок 4.12 Результат редагування профілю користувача

Для додавання нового допису користувач повинен перейти на головну сторінку та натиснути на форму додавання нового допису. Після буде

відображена форма в якій користувач повинен обрати зображення або відео з комп'ютеру або зробити його зі своєї веб камери, та додати опис до зображення.

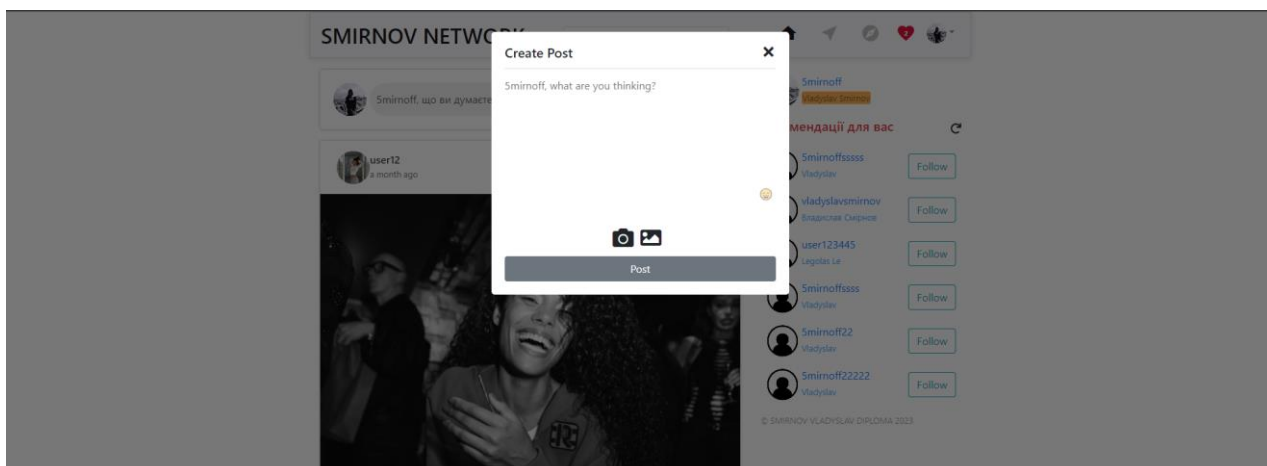


Рисунок 4.13 Форма додавання допису

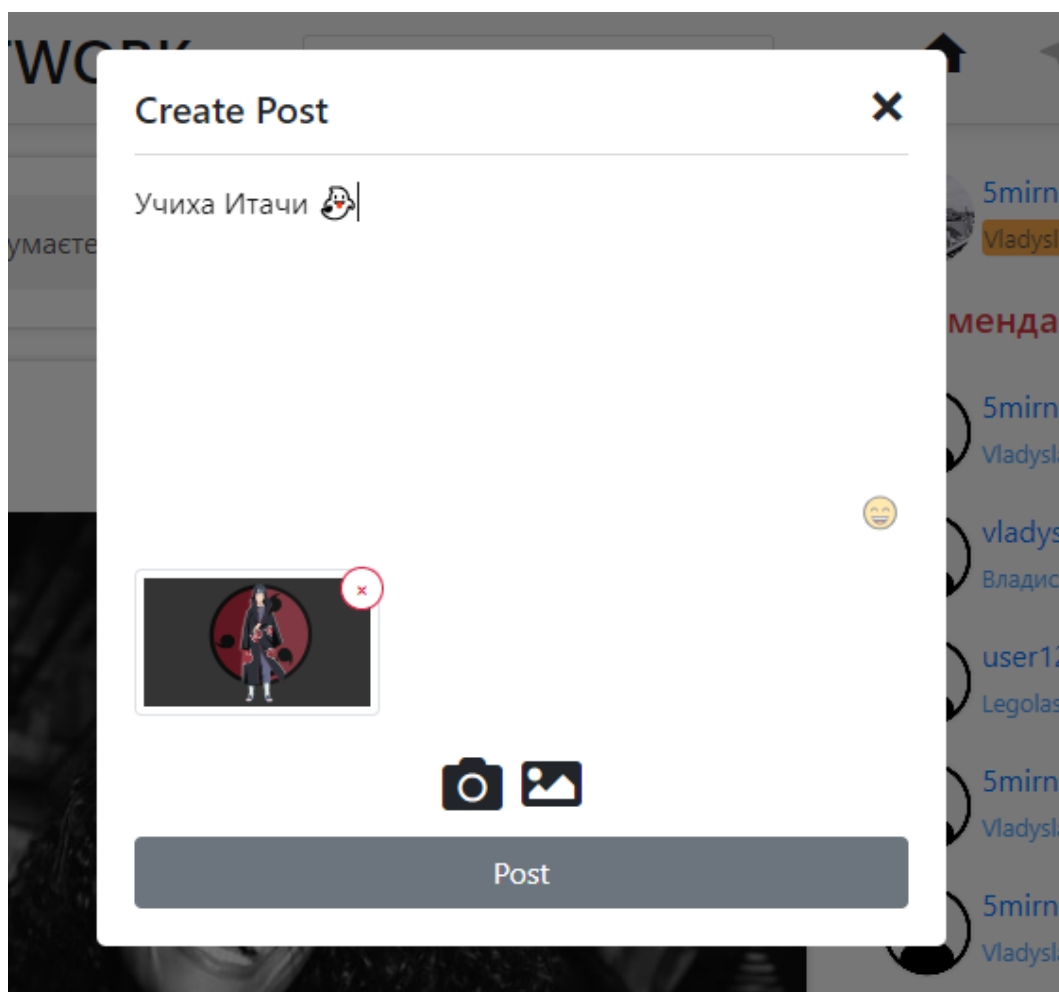


Рисунок 4.14 Заповнена форма додавання допису

Після натискання на кнопку «Пост», допис користувача буде завантажено та він буде відображений як в профілі користувача так і на головній сторінці веб-додатку.

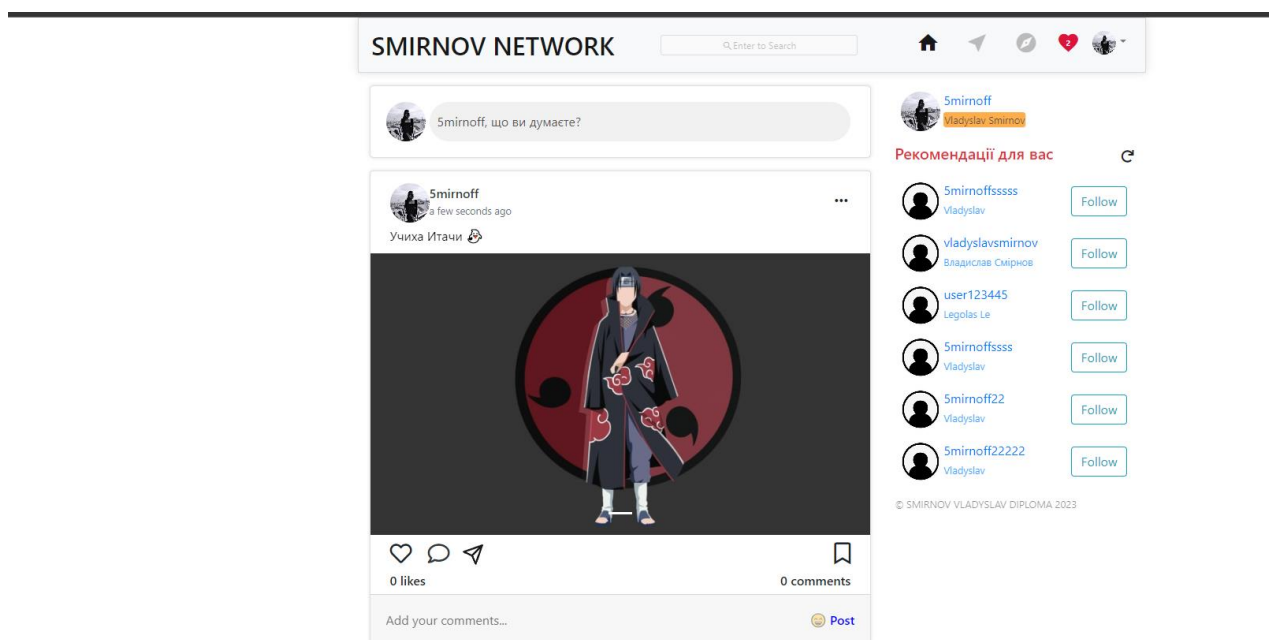


Рисунок 4.15 Новий допис на головній сторінці

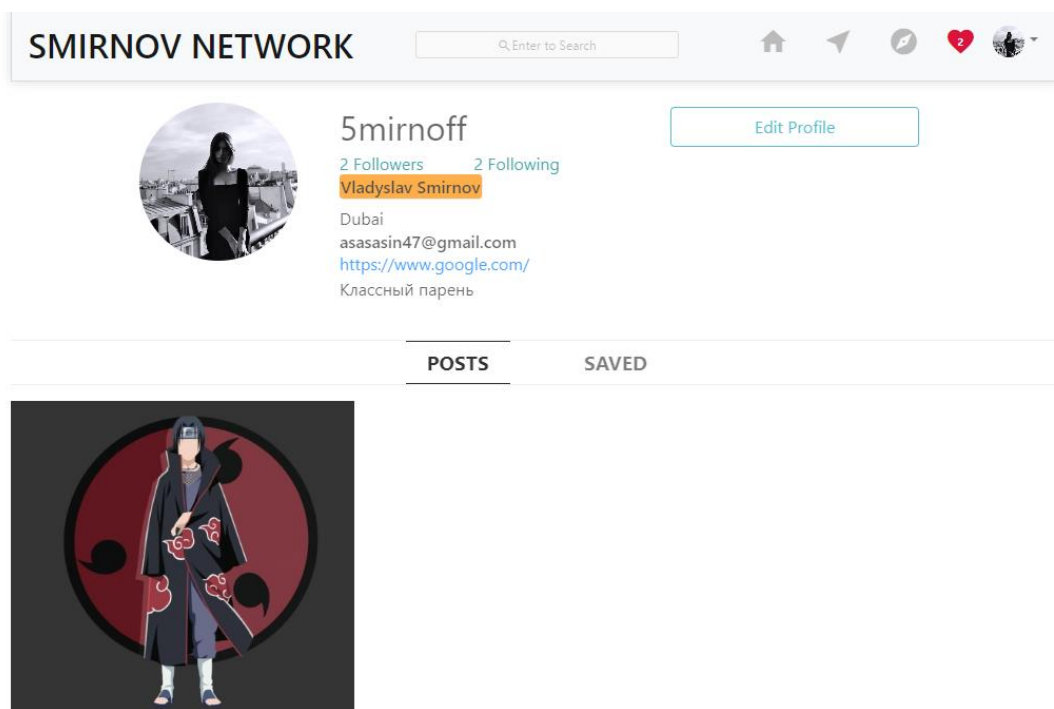


Рисунок 4.16 Новий допис в профілі користувача

Користувач також має можливість редагувати, видалити або скопіювати посилання на допис.

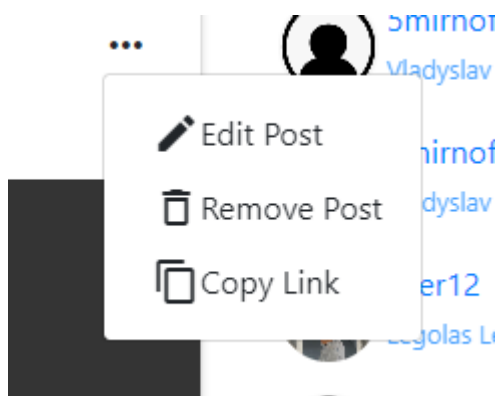


Рисунок 4.17 Взаємодія з дописом

Також в веб-додатку існує система сповіщень, в ній будуть зображені сповіщення про лайки, коментарі на дописи, нові підписки на ваш профіль, якщо людина на яку ви підписані також додасть до свого профілю новий допис.

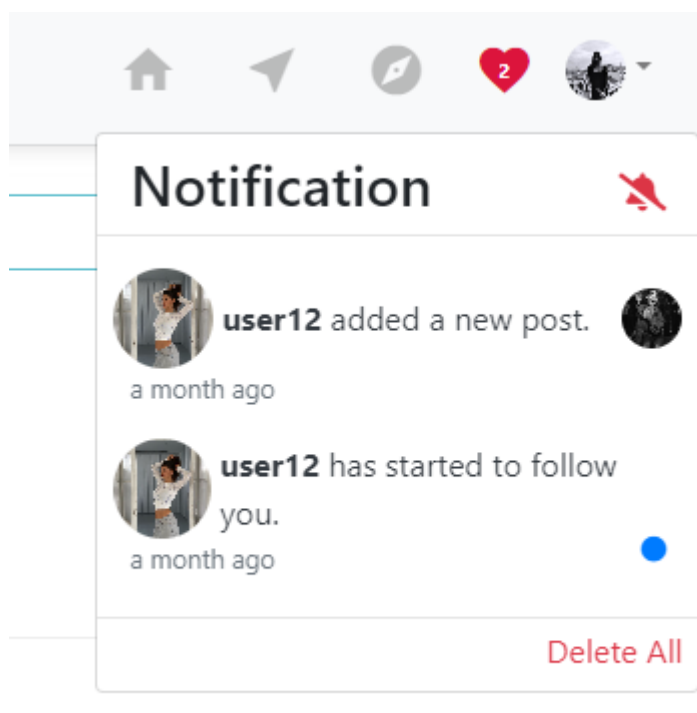


Рисунок 4.18 Сповіщення користувача

Користувач після того як прочитає сповіщення може їх спокійно видалити натиснувши на кнопку «Видалити всі».

Після цього всі сповіщення будуть видалені, а форма сповіщень очищена.

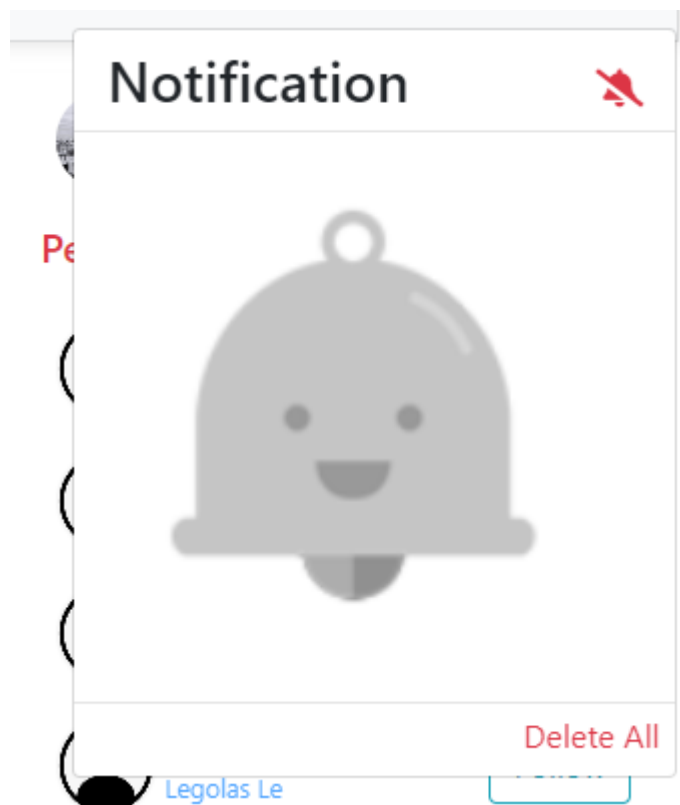


Рисунок 4.19 Очищена форма сповіщень

Як і в більшості сучасних соціальних мереж було розроблено чат, який включає в себе обмін повідомленнями, зображеннями, а також і дзвінки, як аудіо так і відео.

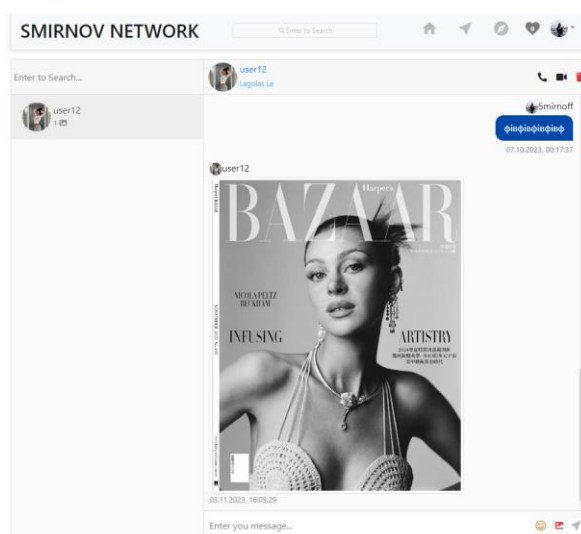


Рисунок 4.20 Чат між двома користувачами

Для здійснення дзвінка потрібно натиснути на кнопку телефона або камери в NavBar зверху.

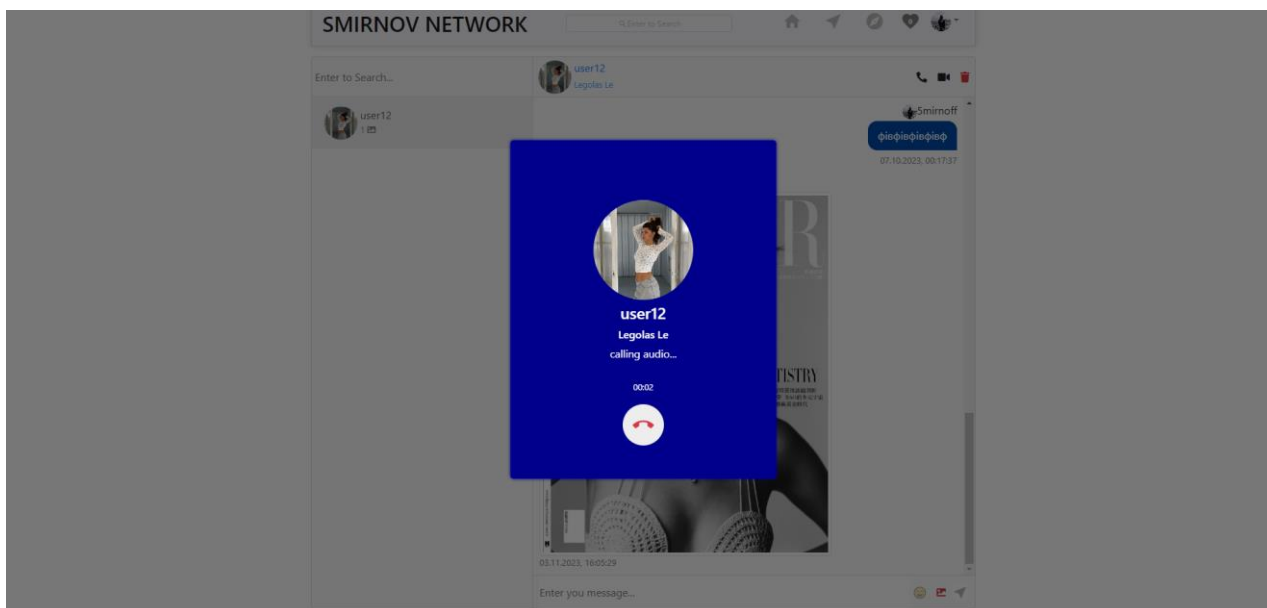


Рисунок 4.21 Здійснення дзвінка

Після успішного дзвінка в чаті буде відображено його, а також скільки він продовжувався.

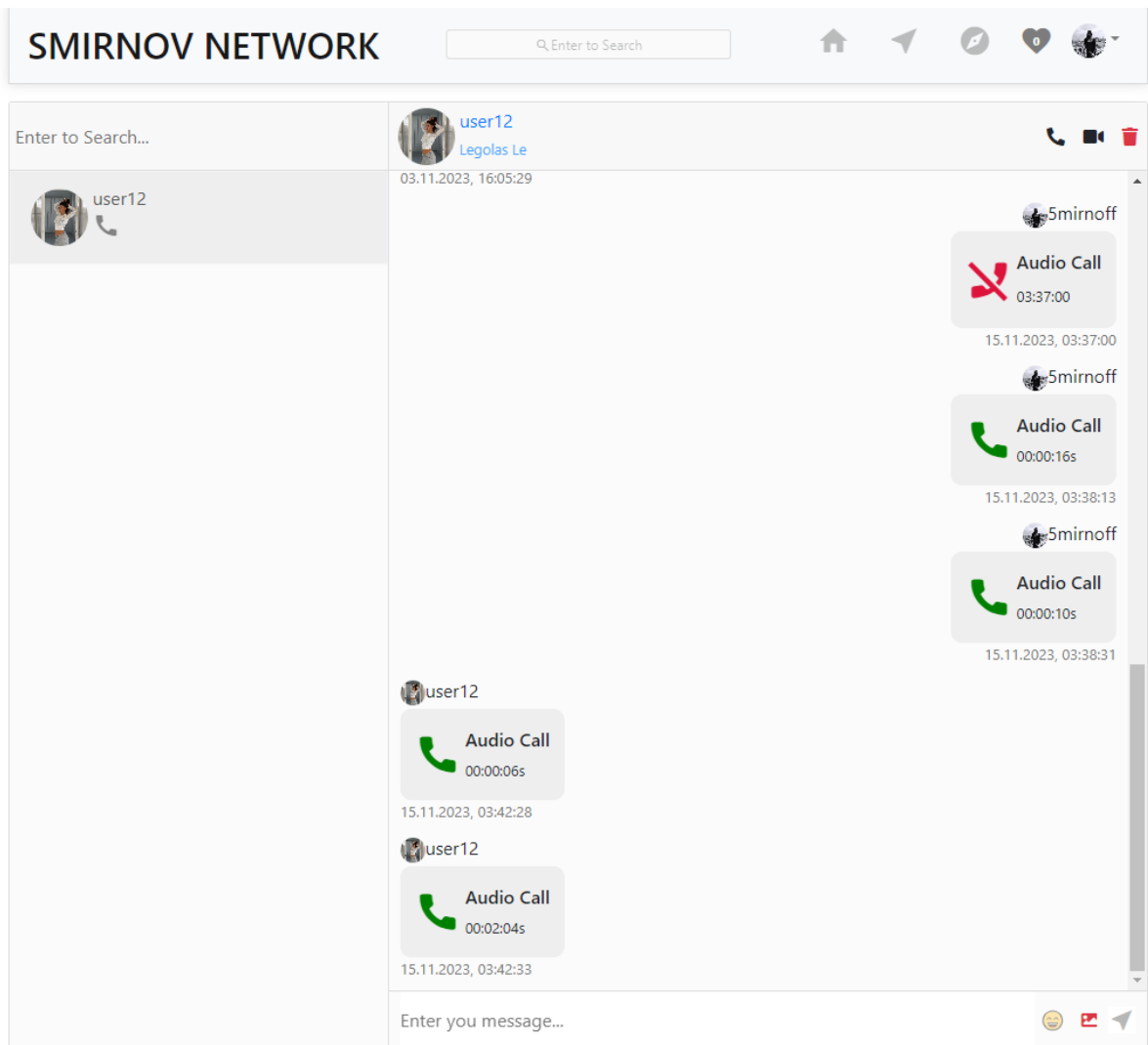


Рисунок 4.22 Відображення дзвінків в чаті

Також було реалізовано видалення чата для користувача, для цього потрібно натиснути на кнопку «Смітник» та чат буде видалено одразу у обох користувачів.

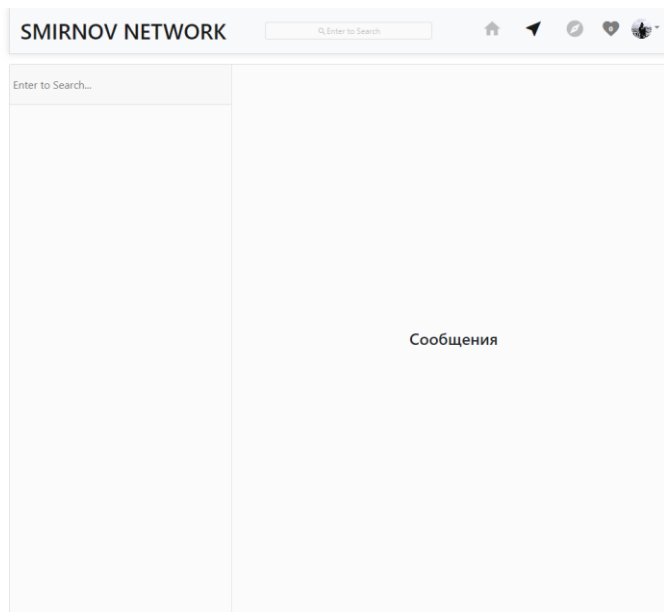


Рисунок 4.23 Результат видалення чату

Було реалізовано пошук користувачів за ніком.

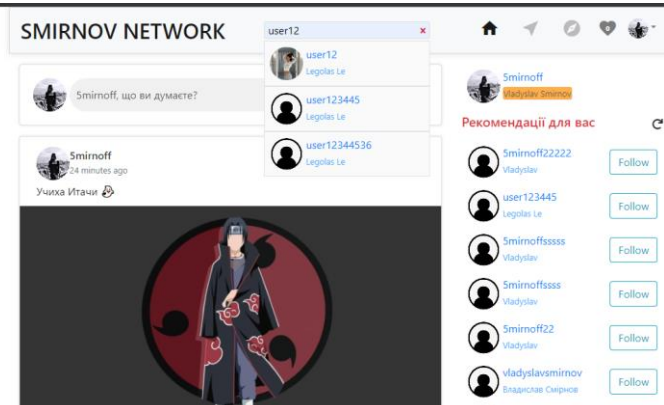




Рисунок 4.23 Пошук Користувачів

Було реалізовано систему лайків та коментарів на дописи.


 **5mirnoff**
26 minutes ago


Учиха Итаци 🤖



👍 🗨️ 📍 📌

1 likes 2 comments

 **5mirnoff**
Ваууу
a few seconds ago 0 likes reply

 **5mirnoff**
класс
a few seconds ago 0 likes reply

Add your comments... 🗨️ Post

Рисунок 4.24 Взаємодія з дописом

Була реалізована сторінка «Цікаво» на якій будуть відображені дописи користувачів на яких ви не підписані.

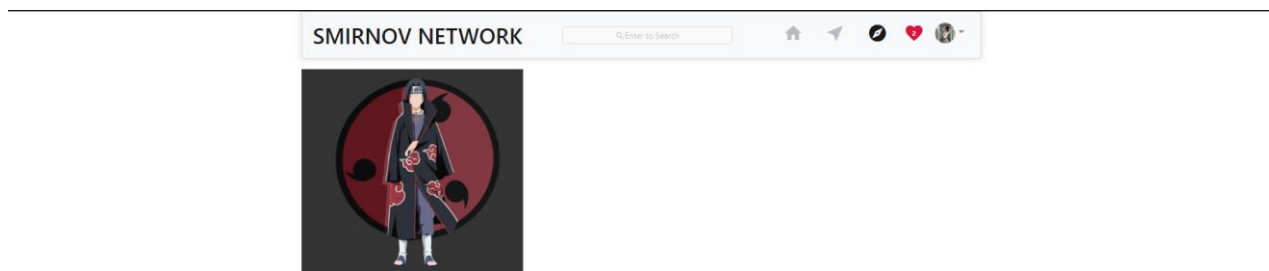


Рисунок 4.25 Сторінка «Цікаво»

Також була реалізована система збереження дописів інших користувачів, для їх збереження потрібно натиснути на цю кнопку.



Рисунок 4.26 Кнопка збереження допису

Після збереження допис буде відображено в профілі користувача в вкладці збережене.

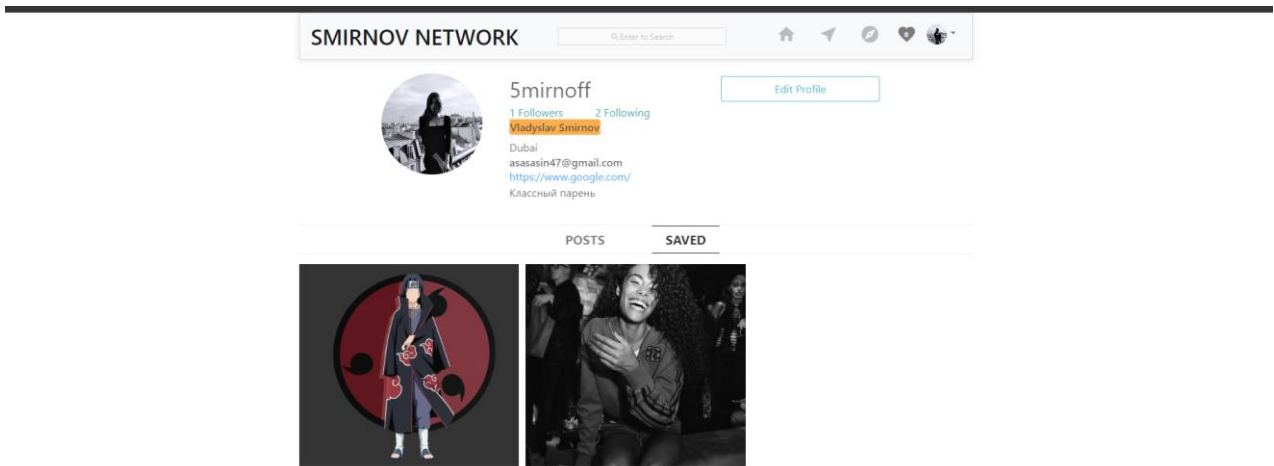


Рисунок 4.27 Збережені дописи

ВИСНОВОК ДО РОЗДІЛУ 4

У четвертому розділі АВР виконано тестування готового веб-додатку, були представлені ключові функції та можливості соціальної мережі. Результатом цього етапу роботи став високофункціональний веб-додаток, який проходив важливі випробування щодо його інтерфейсу та основних функціональних можливостей.

Під час тестування інтерфейсу було перевірено його зручність та доступність для користувача. Ретельно протестовано такі важливі функції, як додавання нових дописів, функціонал чату та дзвінків, редагування профілю, пошук користувачів, можливість вираження симпатій (лайки) та залишення коментарів.

Окрім цього, проведено тестування механізмів реєстрації та авторизації, щоб переконатися у безпеки та надійності цих процесів. Всі виявлені під час тестування помилки та недоліки були усунуті, забезпечуючи стабільну та ефективну роботу веб-додатку.

В результаті цього розділу було досягнуто високого рівня функціональності та якості веб-додатку. Готовий продукт відповідає поставленим завданням та вимогам, а його інтерфейс та функціонал готові до використання користувачами.

ВИСНОВОК

У даній АВР було розроблено веб-додаток соціальної мережі. Усі чотири розділи АВР представляють собою важливі кроки у створенні веб-додатку соціальної мережі. Робота розпочалася з вивчення предметної області, де визначено ключові аспекти соціальних мереж, їхню роль у сучасному інформаційному просторі та різноманітність переваг та недоліків, пов'язаних із їх використанням. З урахуванням цього аналізу була визначена мета та поставлені завдання для подальшої розробки соціальної мережі. Важливим моментом став порівняльний аналіз існуючих рішень, що сприяв визначенню конкурентних переваг розроблюваної соціальної мережі.

Другий розділ присвячений глибокому аналізу та проектуванню веб-додатку соціальної мережі. Використання структурованої методології та UML-діаграм дозволило чітко визначити цілі та завдання, побудувати ефективну концептуальну модель, а також врахувати важливі аспекти системи, такі як взаємодії користувачів та структура об'єктів.

Третій розділ розкриває важливий вибір технічних рішень, включаючи середовище розробки, мову програмування, платформу, бібліотеки та базу даних. Кожен з цих виборів був обґрунтований врахуванням вимог щодо ефективності, зручності розробки та надійності.

У четвертому розділі проведено тестування готового веб-додатку, де визначено його високий рівень функціональності та якості. Результати тестування дозволили виявити та усунути всі помилки, гарантуючи стабільну та безперебійну роботу соціальної мережі.

АВР успішно вирішила поставлені завдання, надаючи детальний аналіз, ефективно проектування та високоякісну реалізацію соціальної мережі. Вивчення предметної області, розробка веб-додатку та тестування дозволили створити продукт, який відповідає вимогам та готовий задовольняти потреби користувачів. Робота може слугувати цінним внеском у сферу розробки соціальних мереж та їхнього подальшого розвитку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Соціальні мережі: поняття, історія виникнення. [Електронний ресурс]. – Режим доступу до ресурсу: <https://zounb.zp.ua/resource/zaporizkyy-kray/zaporizhzhya-bibliotechne/fahova-osvita/socialni-merezhi-piv#q2>
2. Що таке соціальні мережі? Види, класифікація, безпека. [Електронний ресурс]. – Режим доступу до ресурсу: <https://futurenow.com.ua/shho-take-sotsialni-merezhi-vydy-klasyfikatsiya-bezpeka/>
3. Найпопулярніші соціальні мережі у світі 2023 [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.proidei.com/social-media-statistics-0708/>
4. Все про Instagram – повний посібник. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.ranktracker.com/uk/blog/all-about-instagram-the-ultimate-guide-seo-facts-stats/>
5. The Facebook Effect / David Kirkpatrick, 2011 – 384 с.
6. Що таке Фейсбук: навіщо він потрібен і як ним користуватися. Електронний ресурс]. – Режим доступу до ресурсу: <https://107.com.ua/blog/sho-take-feisbyk-nav%D1%96sho-v%D1%96n-potr%D1%96ben-%D1%96iak-nim-koristyvatisia/>
7. Що таке Твіттер, для чого він потрібен: Twitter як соціальна мережа. Електронний ресурс]. – Режим доступу до ресурсу: <http://smartandyoung.com.ua/shho-take-twitter-dlja-chogo-vin-potriben-twitter>
8. Goals!: How to Get Everything You Want -- Faster Than You Ever Thought Possible / Brian Tracy, 2003 – 304 с.
9. Дерево цілей: переваги і побудова. [Електронний ресурс]. – Режим доступу до ресурсу: <https://pyrogiv.kiev.ua/shho-take-derevo-cilej/>
10. Modern Systems Analysis and Design, Global Edition / Joseph S Valacich, Joey F. George, 2015 – 544 с.
11. Знайомство з контекстною діаграмою. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.mindonmap.com/uk/blog/context-diagram/>

12. Conceptual Modeling: Foundations and Applications / Eric Yu, Paolo Giorgini, Alex T. Borgida, Vinay Chaudhri, 2009 – 530 с.
13. Поняття ER-моделі. Поняття сутності (entity). Атрибути. Види атрибутів. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.bestprog.net/uk/2019/01/24/the-concept-of-er-model-the-concept-of-essence-and-communication-attributes-attribute-types-ua/>
14. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development / Craig Larman, 2004 – 736 с.
15. Діаграма прецедентів. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.wikiwand.com/uk/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D1%96%D0%B2
16. Unified Modeling Language User Guide, The (2nd Edition) / Grady Booch, James Rumbaugh, Ivar Jacobson, 2017 – 504 с.
17. Діаграми класів. [Електронний ресурс]. – Режим доступу до ресурсу: <https://ua5.org/oop/392-diagrami-klasiv.html>
18. UML 2.0 in Action: A Project-Based Tutorial / Patrick Graessle, Mike Pilone, 2005 – 248 с.
19. Діаграми послідовності. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.maxzosim.com/sequence-diagrams/>
20. Object-Oriented Modeling and Design with UML / Michael Blaha, James R. Rumbaugh, 2005 – 496 с.
21. What is activity diagram? [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>
22. Learn to code with Visual Studio Code 2023. [Електронний ресурс]. – Режим доступу: <https://code.visualstudio.com/learn>.
23. HTML підручник. [Електронний ресурс]. – Режим доступу: <https://w3schoolsua.github.io/html/index.html#gsc.tab=0>

24. CSS підручник. [Електронний ресурс]. – Режим доступу: <https://w3schoolsua.github.io/css/index.html#gsc.tab=0>.
25. Learning JavaScript: JavaScript Essentials for Modern Application Development 3rd Edition / Ethan Brown. – К.: O'Reilly Media, 2016
26. Node.js v20.9.0 documentation 2023. [Електронний ресурс]. – Режим доступу: <https://nodejs.org/docs/latest-v20.x/api/>
27. Most popular frontend frameworks for web development in 2022. [Електронний ресурс]. – Режим доступу до ресурсу: <https://twelve devs.com/blog/most-popular-frontend-frameworks-for-web-development-in-2022>
28. Переваги та недоліки React.JS. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.affiliatebay.net/uk/advantages-and-disadvantages-of-react-js/>
29. React Hooks. [Електронний ресурс] – Режим доступу до ресурсу: <https://react-redux.js.org/api/hooks>
30. Redux documentation 2023. [Електронний ресурс]. – Режим доступу: <https://rajdee.gitbooks.io/redux-in-russian/content/docs/basics/Reducers.html>.
31. Redux essentials. [Електронний ресурс]. – Режим доступу до ресурсу: <https://redux.js.org/tutorials/essentials/part-1-overview-concepts>
32. Library API References for jsonwebtoken library. 2023. [Електронний ресурс]. – Режим доступу: <https://www.npmjs.com/package/jsonwebtoken>.
33. Bcrypt API Reference. 2022. [Електронний ресурс]. – Режим доступу: <https://www.npmjs.com/package/bcrypt>.
34. Socket.IO Docs. 2023. [Електронний ресурс]. – Режим доступу: <https://socket.io/docs/v4/>
35. MongooseJS API Docs. 2023. [Електронний ресурс]. – Режим доступу: <https://mongoosejs.com/docs/api.html>

ДОДАТОК А – ІНФОРМАЦІЙНІ СЛАЙДИ

Київський національний університет будівництва і архітектури
Автоматизації і інформаційних технологій
Інформаційних технологій

Атестаційна випускна робота
на здобуття освітнього ступеню «магістр» на тему:
«Розробка веб-додатку соціальної мережі»

Виконав
Студент групи КНМ-22
Спеціальності 122 «Комп'ютерні науки»
Смірнов В.Р.
Керівник к.т.н., доц. Поплавський О.А.
Рецензент к.т.н., доц. Горда О.В.

Київ 2023

Вступ

В сучасному світі, інформаційні технології та мережа Інтернет перетинають усі сфери нашого життя, змінюючи їх завжди й назавжди. Один з найбільш суттєвих внесків в цю цифрову революцію полягає в розробці веб-додатків, зокрема соціальних мереж. Сучасні соціальні мережі стали не лише платформою для зв'язку та обміну інформацією, але й важливим фактором формування культури, взаємозв'язків і впливу на суспільство.

Ця АВР присвячена розробці веб-додатку соціальної мережі, і вона ставить перед собою завдання дослідити, розробити та реалізувати платформу, яка відповідає сучасним потребам користувачів в онлайн-спільноті.

Аналіз предметної області

Соціальна мережа – це веб-платформа або програмне забезпечення, яке дозволяє користувачам створювати особисті профілі, обмінюватись інформацією, спілкуватися з іншими користувачами та будувати віртуальні зв'язки.

Переваги соціальних мереж:

- Спілкування
- Знаходження інформації
- Бізнес та маркетинг
- Самовираження
- Розваги та дозвілля

Недоліки:

- Приватність та безпека
- Відповідальність за контент
- Залежність
- Спам та реклама
- Кіберзлочинність

Постановка задачі

Необхідно розробити веб-додаток соціальної мережі з наступною функціональністю:

- створення дописів, їх редагування та видалення;
- пошук користувачів;
- перегляд дописів користувачів, їх коментування та лайки;
- рекомендації для користувачів;
- профіль користувача, та можливість його редагування;
- створення системи підписок між користувачами;
- створення онлайн чату в реальному часі між користувачами;
- створення можливості аудіо та відео дзвінків між користувачами;
- створення системи сповіщень.

Аналіз існуючих рішень

Instagram

Instagram – це популярна соціальна мережа, яка спеціалізується на обміні фотографіями та короткими відеозаписами. Основний функціонал Instagram включає в себе можливість завантаження фотографій та відео, їх обробку за допомогою різних фільтрів та інструментів, а також спільне спілкування та обмін враженнями.

Facebook

Facebook - це одна з найпопулярніших соціальних мереж у світі, яка дозволяє користувачам спілкуватися, обмінюватися відомостями та контентом, створювати профілі та спільноти і багато іншого.

Twitter

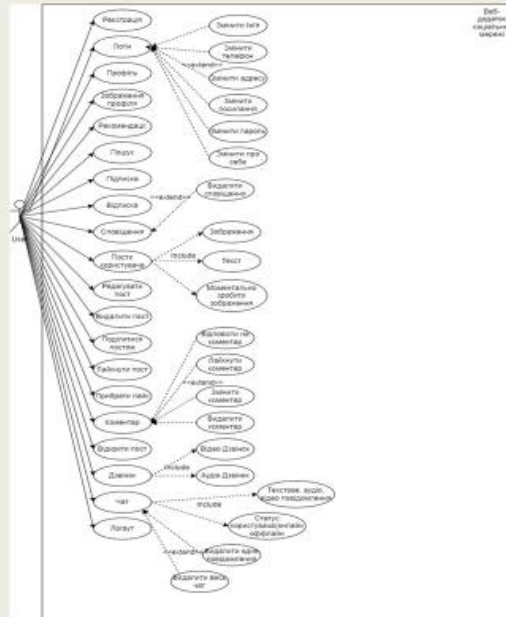
Twitter – це соціальна мережа та онлайн-платформа для мікроблогінгу, яка дозволяє користувачам публікувати короткі повідомлення, відомі як «твіти». Користувачі можуть ділитися новинами, враженнями, посиланнями та іншим контентом.

Аналіз існуючих рішень

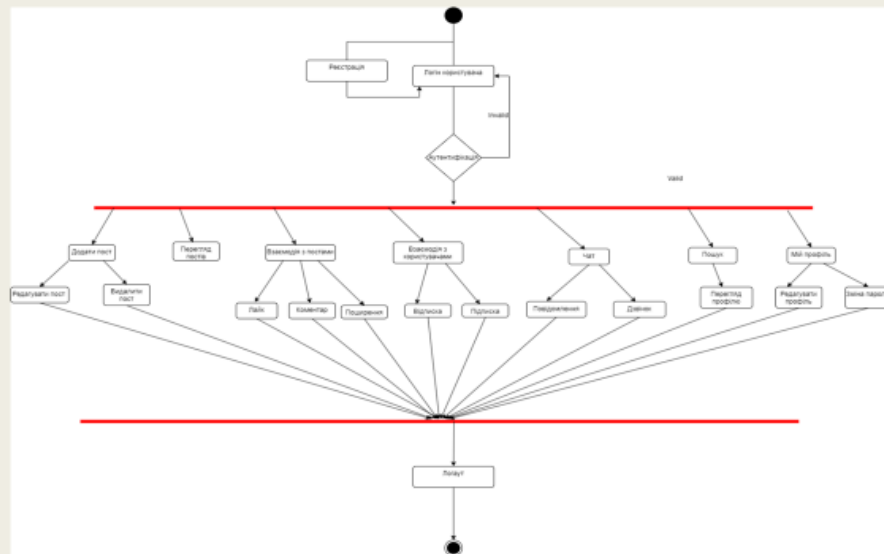
Порівняльна таблиця існуючих та розробленої системи

	Instagram	Facebook	Twitter	Smirnov Network
Регістрація, вхід та вихід	+	+	+	+
Валювання введених даних	+	+	+	+
Перевірка на роботу	+	+	+	-
Створення, редагування, видалення дописів	+	+	+	+
Підписка на дописи	+	+	+	+
Чатинг між користувачами	+	+	+	+
Аудіо та відеозаписи	+	+	-	+
Лайки на дописи	+	+	+	+
Коментарі	+	+	+	+
Перегляд профілю	+	+	+	+
Редагування профілю	+	+	+	+
Мобільний додаток	+	+	+	-

Діаграма варіантів використання



Діаграма діяльності



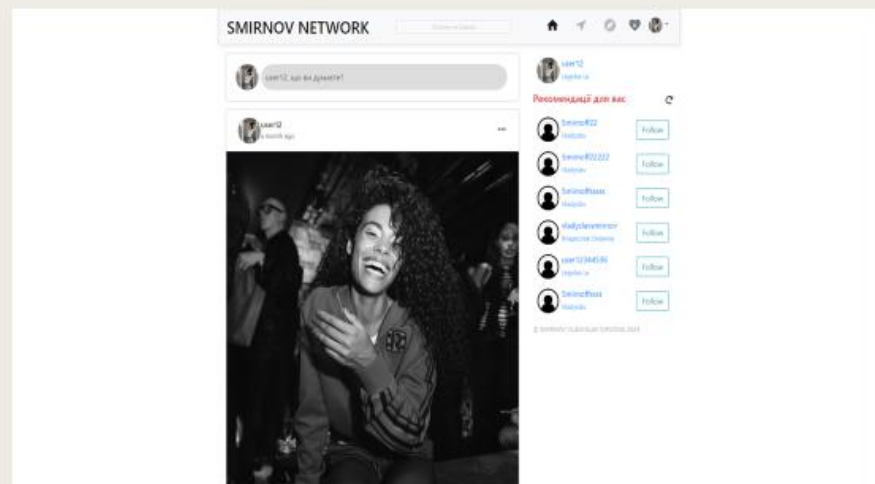
Вибір засобів розробки

Для розробки веб-додатку соціальної мережі було використано такі технології розробки:

- Мова програмування JavaScript;
- Програмна платформа Node.js;
- Фреймворк Express.js;
- Бібліотека React;
- Бібліотека Redux;
- База даних MongoDB;
- Бібліотека JsonWebToken;
- Бібліотека Bcrypt;
- Бібліотека Socket.io.

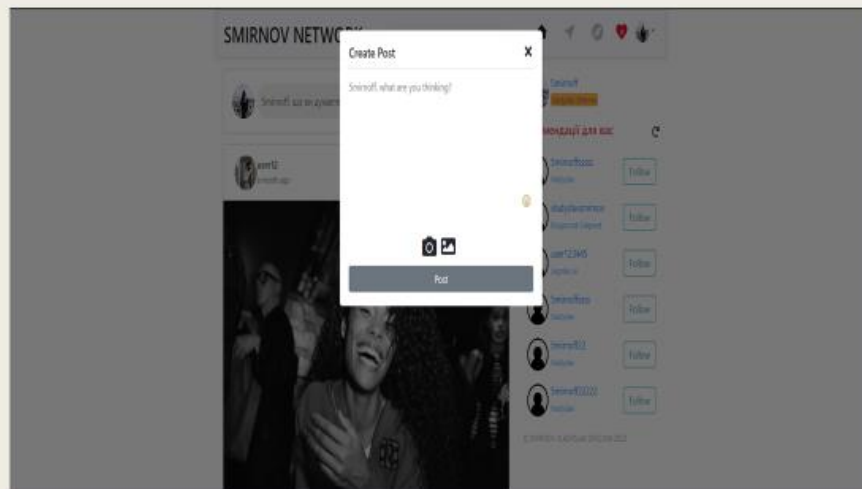
Результати розробки

Головна сторінка веб-додатку



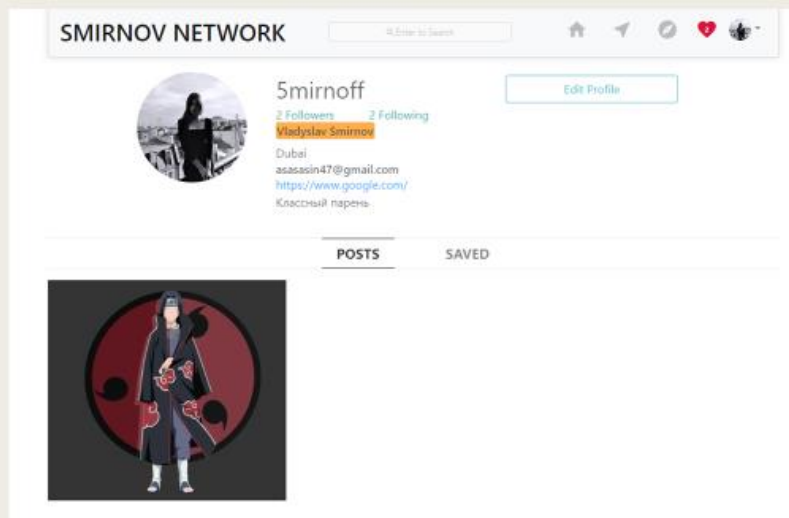
Результати розробки

Форма додавання нового допису



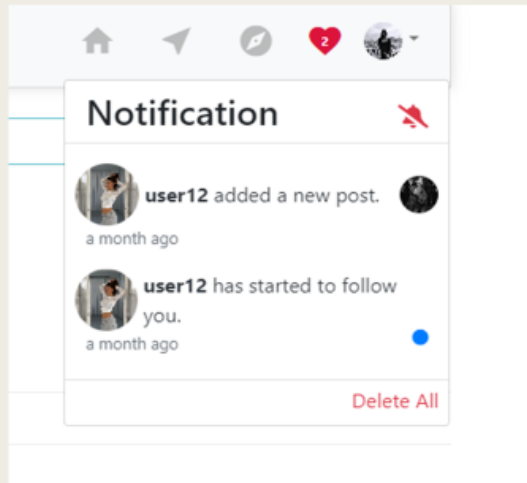
Результати розробки

Сторінка профілю користувача



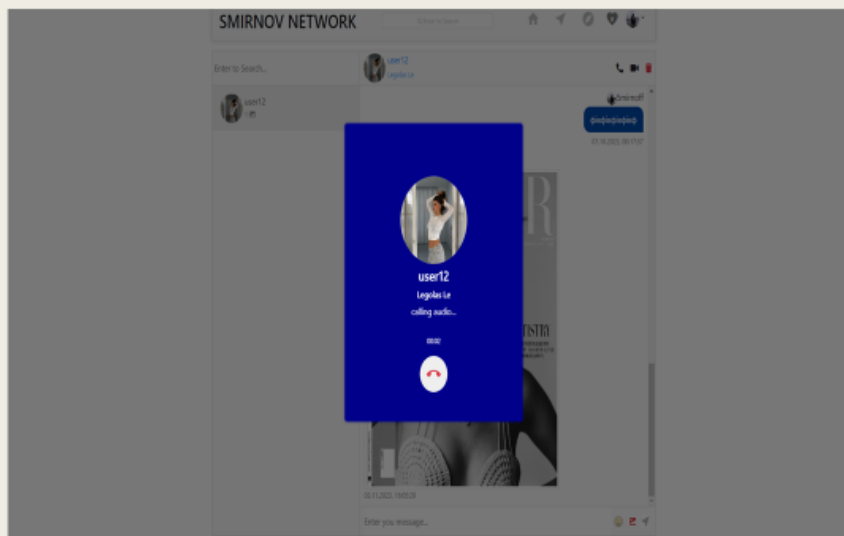
Результати розробки

Модуль сповіщень користувача



Результати розробки

Здійснення дзвінка



Висновок

АВР успішно вирішила поставлені завдання, надаючи детальний аналіз, ефективне проектування та високоякісну реалізацію соціальної мережі. Вивчення предметної області, розробка веб-додатку та тестування дозволили створити продукт, який відповідає вимогам та готовий задовольняти потреби користувачів. Робота може слугувати цінним внеском у сферу розробки соціальних мереж та їхнього подальшого розвитку.

Дякую за увагу!

