

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Бакалаврська дипломна робота

на тему: «Розробка інтерактивної комунікаційної системи передачі
повідомлень»

Виконав: студент 4 курсу

групи ЗП-20Б

спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Качковецький А. І.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Рейда О.М.

(прізвище та ініціали)

Рецензент: д.т.н., професор Мартинюк Т.Б.

(прізвище та ініціали)

Допущено до захисту

Зав. кафедри ПЗ

«10» 06 2024 р.

ВНТУ – 2024

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

О. Н. Романюк

« 12 » березня 2024 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Качковецькому Андрію Ігоровичу

1. Тема роботи – «Розробка інтерактивної комунікаційної системи передачі повідомлень».

Керівник роботи: Рейда Олександр Миколайович, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від 11 березня 2024 р. № 80.

2. Строк подання студентом роботи 3.червня 2024.

3. Вихідні дані до роботи: середовище розробки Visual Studio Code, мова розробки PHP, операційна система - Windows 10, система управління вмістом WordPress, базові алгоритми.

4. Зміст текстової частини: комп'ютериний застосунок; модель розробки каскадна; система управління змістом – WordPress; вхідні дані: персональні дані входу користувачів; вихідні дані: відображення публічних та приватних повідомлень користувачів; засоби розробки – MS Visual Studio Code; мова програмування – PHP.

5. Перелік ілюстративного матеріалу: титульний слайд; актуальність теми; мета, об'єкт та предмет дослідження; новизна і практична цінність одержаних результатів; порівняльний аналіз аналогів; структурна схема; діаграма послідовності; інтерфейс додатка, вікна додатку, апробації та публікації, висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Рейда О.М., к.т.н., доцент кафедри ПЗ	13.03.24	03.06.24

7. Дата видачі завдання 13 березня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ п\п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз інтерактивних системи передачі повідомлень	13.03.24 – 22.03.24	виконано
2	Розробка інтерфейсу та моделей системи	13.03.24 – 22.03.24	виконано
3	Розробка програмних модулів реалізації вебдодатку	25.03.24 – 19.04.24	виконано
4	Тестування вебдодатку інтерактивної системи передачі повідомлень	22.04.24 – 17.05.24	виконано
5	Оформлення матеріалів до захисту БДР	20.05.24 – 03.06.24	виконано

Студент А. І. Качковецький
(підпис) (ініціали та прізвище)

Керівник бакалаврської дипломної роботи О. М. Рейда
(підпис) (ініціали та прізвище)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 57 сторінок формату А4, на яких є 34 рисунків, 6 таблиць, список використаних джерел містить 20 найменувань.

У бакалаврській дипломній роботі проведено аналіз систем передачі повідомлень. Було проведено аналіз аналогів, визначено їх переваги та недоліки, доведено доцільність розробки власного програмного застосунку.

Виконано обґрунтування вибору мови та середовища програмування, а також технологій, які були використані під час розробки веб-застосунку застосунку. Розроблено програмний продукт, який являє собою веб-сайт.

Робота реалізована за допомогою мови програмування PHP. В якості системи керування змістом було обрано WordPress.

Отримані в бакалаврській дипломній роботі результати можна використати для інтерактивної передачі повідомлень.

Ключові слова: повідомлення, інтерактив, веб-застосунок.

Abstracts

The bachelor thesis consists of 57 pages of A4 format, on which there are 34 figures, 6 tables, the list of used sources contains 20 names.

An analysis of message transmission systems was carried out in the bachelor thesis. An analysis of analogues was carried out, their advantages and disadvantages were determined, and the expediency of developing one's own software application was proven.

The justification of the choice of the language and programming environment, as well as the technologies that were used during the development of the web application of the application, is performed. A software product, which is a website, has been developed.

The work is implemented using the PHP programming language. WordPress was chosen as the content management system.

The results obtained in the bachelor thesis can be used for interactive transmission of messages.

Keywords: message, interactive, web application.

ЗМІСТ

ВСТУП	4
1.1 Аналіз стану вебсистем з використанням програмних засобів передачі повідомлень.....	7
1.2 Порівняльний аналіз аналогів.....	9
1.3 Аналіз методів розв’язання задачі.....	15
1.4 Постановка задач проекту	16
1.5 Висновки	16
2 РОЗРОБКА ІНТЕРФЕЙСУ НА МОДЕЛЕЙ СИСТЕМИ	18
2.1 Аналіз вхідних даних системи.....	18
2.2 Розробка інтерфейсу програмного додатку.....	18
2.3 Розробка моделі системи.....	25
2.4 Розробка модулів для роботи системи.....	26
2.5 Висновки	27
3.РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ В ВЕБДОДАТКУ	28
3.1 Варіантний аналіз і обґрунтування вибору методів та засобів для реалізації програмного забезпечення	28
3.2 Розробка головного екрану та навігації.....	35
3.3 Розробка модуля передачі повідомлень.....	40
3.4 Розробка модуля інтегрування штучного інтелекту	42
3.5 Висновки	45
4 ТЕСТУВАННЯ ПРОГРАМИ	46
4.1 Тестування системи	46
4.2 Розробка інструкції користувача.....	54
4.3 Висновки	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТКИ.....	62

Додаток А – ТЕХНІЧНЕ ЗАВДАННЯ	63
Додаток Б – ПРОТОКОЛ ПЕРЕВІРКИ НА ПЛАГІАТ	66
Додаток В – ЛІСТИНГ ПРОГРАМИ.....	67
Додаток Г – ІЛЮСТРАТИВНИЙ МАТЕРІАЛ.....	83

ВСТУП

Обґрунтування вибору теми дослідження. В умовах стрімкого розвитку інформаційних технологій освітні заклади потребують ефективних засобів комунікації, які дозволяють швидко та надійно передавати інформацію між учасниками навчального процесу. Сучасні виклики, з якими стикаються школи, коледжі та університети, включають необхідність оперативного обміну інформацією, інтерактивної взаємодії між учнями, вчителями та адміністрацією, а також підвищення рівня залученості батьків у навчальний процес.

На сьогоднішній день існує ряд потреб для таких систем:

1. Потреба в оперативній комунікації. Традиційні методи передачі інформації, такі як паперові повідомлення, оголошення на дошках або електронна пошта, часто виявляються недостатньо ефективними в умовах динамічного навчального процесу. Важливі повідомлення можуть бути втрачені або прочитані із запізненням, що призводить до непорозумінь та збоїв у роботі закладу. Інтерактивна комунікаційна система, яка забезпечує миттєву передачу повідомлень, допоможе уникнути цих проблем та підвищити ефективність управління навчальним процесом[2].

2. Інтерактивна взаємодія та зворотній зв'язок. Одним з ключових аспектів сучасної освіти є інтерактивність та можливість зворотного зв'язку. Учні, вчителі та батьки потребують платформи, яка дозволяє легко взаємодіяти, обговорювати важливі питання, брати участь в опитуваннях та голосуваннях. Інтерактивна комунікаційна система сприятиме створенню більш відкритого та залученого освітнього середовища, де кожен учасник може швидко отримати відповіді на свої питання та надати свій внесок у розвиток навчального процесу.

3. Підтримка дистанційного навчання Пандемія COVID-19 показала важливість та необхідність дистанційного навчання. Для забезпечення його ефективності освітні заклади потребують надійних засобів комунікації, які дозволяють підтримувати зв'язок між усіма учасниками навчального процесу незалежно від їхнього місцезнаходження. Інтерактивна комунікаційна система,

що включає функції відеоконференцій, чатів та інтеграції з освітніми платформами, стане невід'ємною частиною сучасного навчального процесу, забезпечуючи безперервність та якість освіти[3].

4. **Персоналізований підхід та адаптація до потреб користувачів** Кожен учасник освітнього процесу має свої унікальні потреби та пріоритети. Інтерактивна комунікаційна система, яка дозволяє налаштовувати сповіщення та повідомлення відповідно до індивідуальних потреб, сприятиме більш ефективному обміну інформацією та підвищенню задоволеності користувачів. Така система дозволить учням, вчителям та батькам отримувати тільки релевантну для них інформацію, що знизить інформаційне перевантаження та підвищить ефективність комунікації.

Тому реалізація інтерактивної системи передачі повідомлень є актуальною розробкою у наш час.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно з планом виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Метою роботи є розробка інтерактивної комунікаційної системи передачі повідомлень для підвищення ефективності та зручності обміну інформацією між користувачами, а також спрямованна на створення сприятливого середовища для взаємодії.

Основними задачами роботи є:

- провести аналіз існуючих методів і засобів систем передачі повідомлень для визначення напрямків підвищення ефективності та зручності обміну інформацією між користувачами;
- запропонувати метод аналізу настроїв та емоційного стану користувачів;
- розробити програмні компоненти та інтерактивну комунікаційну систему передачі повідомлень на основі запропонованих методів;
- провести експериментальні дослідження розроблених засобів.

Об'єктом дослідження є процес розробки інтерактивної системи передачі повідомлень є процес автоматизації процесів у сфері управління персоналом.

Предметом дослідження є методи та засоби розробки інтерактивної системи передачі повідомлень.

Методи дослідження. Під час розробки інтерактивної системи передачі повідомлень використовувалися такі методи: методи лінгвістичного аналізу, методи синтаксичного та семантичного аналізу, методи машинного навчання для класифікації та кластеризації даних, методи візуалізації для аналізу взаємодій, методи моделювання даних, комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

Новизна отриманих результатів. Подальшого розвитку отримав метод аналізу настроїв та емоційного стану користувачів, у якому, на відміну від існуючих, використано штучний інтелект для аналізу емоційного тону повідомлень, що дозволило виявляти настрій користувачів, відповідним чином реагувати, і, як наслідок, зменшити емоційне навантаження при спілкуванні.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає в тому, що на основі отриманих в бакалаврській дипломній роботі теоретичних положеннях було запропоновано сучасні підходи до розробки інтерактивних систем передачі повідомлень, розроблено програмні модулі передачі повідомлень, аналізу емоційного наповнення повідомлень.

Особистий внесок. Всі наукові результати, що викладені у бакалаврській дипломній роботі, були отримані безпосередньо автором роботи.

Апробація матеріалів бакалаврської дипломної роботи. Основні положення бакалаврської дипломної роботи доповідалися та обговорювалися на ЛІІ Науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2024) [1].

Публікації. За тематикою дослідження опубліковано 1 наукову працю у збірниках матеріалів конференцій.

Технічне завдання наведено в додатку А.

1 АНАЛІЗ РОЗВИТКУ ВЕБСИСТЕМ З ВИКОРИСТАННЯМ ПРОГРАМНИХ ЗАСОБІВ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану вебсистем з використанням програмних засобів передачі повідомлень

В останні роки спостерігається стрімкий розвиток інтерактивних систем передачі повідомлень, які використовують ШІ для створення адаптивних і персоналізованих навчальних середовищ. Основними компонентами цих систем є інтерактивні платформи та чат-боти, які забезпечують оперативний обмін інформацією, інтерактивну взаємодію та надання індивідуалізованої допомоги.

Інтерактивні системи передачі повідомлень з використанням ШІ включають у себе різноманітні платформи та інструменти, які підтримують комунікацію між учнями, вчителями та батьками. Ці системи можуть використовувати текстові повідомлення, відеочати, електронні листи та інші форми передачі інформації. ШІ допомагає аналізувати поведінку та потреби учнів, адаптувати навчальні матеріали та забезпечувати зворотний зв'язок у реальному часі. ШІ-чат-боти є ще однією важливою складовою інтерактивних систем передачі повідомлень. Вони можуть миттєво відповідати на запити учнів, допомагати у виконанні завдань та надавати психологічну підтримку.

Технічні аспекти інтерактивних систем передачі повідомлень включають використання технологій машинного навчання та обробки природної мови (NLP). Ці технології дозволяють системам розуміти запити учнів і надавати відповіді в природній формі, що полегшує взаємодію та забезпечує більш природне спілкування. Адаптивні алгоритми, які аналізують успіхи та помилки учнів, їхній стиль навчання та рівень знань, дозволяють надавати релевантні рекомендації та підтримку.

Сучасні інтерактивні системи передачі повідомлень часто інтегруються з іншими освітніми платформами та сервісами для забезпечення безперервного навчального процесу [4]. Це включає інтеграцію з електронними журналами,

платформами для відеоконференцій, системами управління навчальним процесом (LMS) та іншими.

Основними перевагами використання інтерактивних систем передачі повідомлень з ІІІ є індивідуалізація навчання, підвищення доступності освіти та надання необхідної емоційної підтримки учням з особливими освітніми потребами. Такі системи дозволяють створювати індивідуальні навчальні плани, що враховують потреби кожного учня, роблять навчання доступнішим та більш інклюзивним, а також допомагають учням отримувати своєчасну та релевантну підтримку.

Недоліки існуючих рішень:

1. Обмежена функціональність: Більшість існуючих додатків для передачі повідомлень мають обмежений функціонал, що часто не враховує потреб користувачів.
2. Низька зручність використання: Деякі додатки мають складний інтерфейс або неінтуїтивні функції, що ускладнює їх використання для користувачів.
3. Недостатність інтеракції між користувачами. Багато додатків обмежені лише до сповіщень та передачею базових повідомлень, не забезпечуючи повноцінної взаємодії між користувачами.
4. Недостатня безпека та конфіденційність. Деякі додатки можуть не забезпечувати достатнього рівня безпеки для особистих даних користувачів.

На основі виявлених недоліків існуючих рішень та враховуючи потреби спільнот, поставимо наступні задачі для дослідження:

1. Аналіз потреб користувачів. Провести аналіз потреб спільнот та їхніх вимог до функціоналу веб додатку.
2. Розробка функціональних вимог. Сформулювати функціональні вимоги до додатку, враховуючи отримані результати аналізу.
3. Проєктування інтерфейсу. Розробити зручний та інтуїтивно зрозумілий інтерфейс користувача для веб додатку, що відповідає потребам спільнот.

4. Розробка та тестування. Реалізувати програмний засіб у вигляді веб додатку на основі встановлених вимог та провести його тестування на реальних користувачах

5. Оцінка ефективності. Провести оцінку ефективності та корисності веб додатку з точки зору задоволення потреб користувачів та покращення процесу передачі повідомлень.

Ці задачі дозволять нам розробити та реалізувати програмний засіб веб додатку для інтерактивної передачі повідомлень під назвою "akvavit", який відповідає їхнім потребам та сприяє покращенню комунікації в спільноті.

1.2 Порівняльний аналіз аналогів

На ринку існують різні веб додатки, спрямовані на передачу повідомлень в спільноті. Деякі з них вже мають певну популярність, а інші тільки починають набирати обертів на ринку. Давайте розглянемо декілька таких додатків та проведемо їх порівняльний аналіз[5].

Аналіз аналога "QQ".

"QQ" - це китайська соціальна мережа та месенджер, який став дуже популярним в Китаї. Він був розроблений компанією Tencent і вперше випущений у 1999 році. Початково QQ був мобільною версією месенджера для персональних комп'ютерів. (див. рисунок 1.1). Його перевагою доступність базового функціоналу та стабільність додатку. Однак, додаток має застарілий дизайн і не забезпечує ефективну безпеку даних [6].

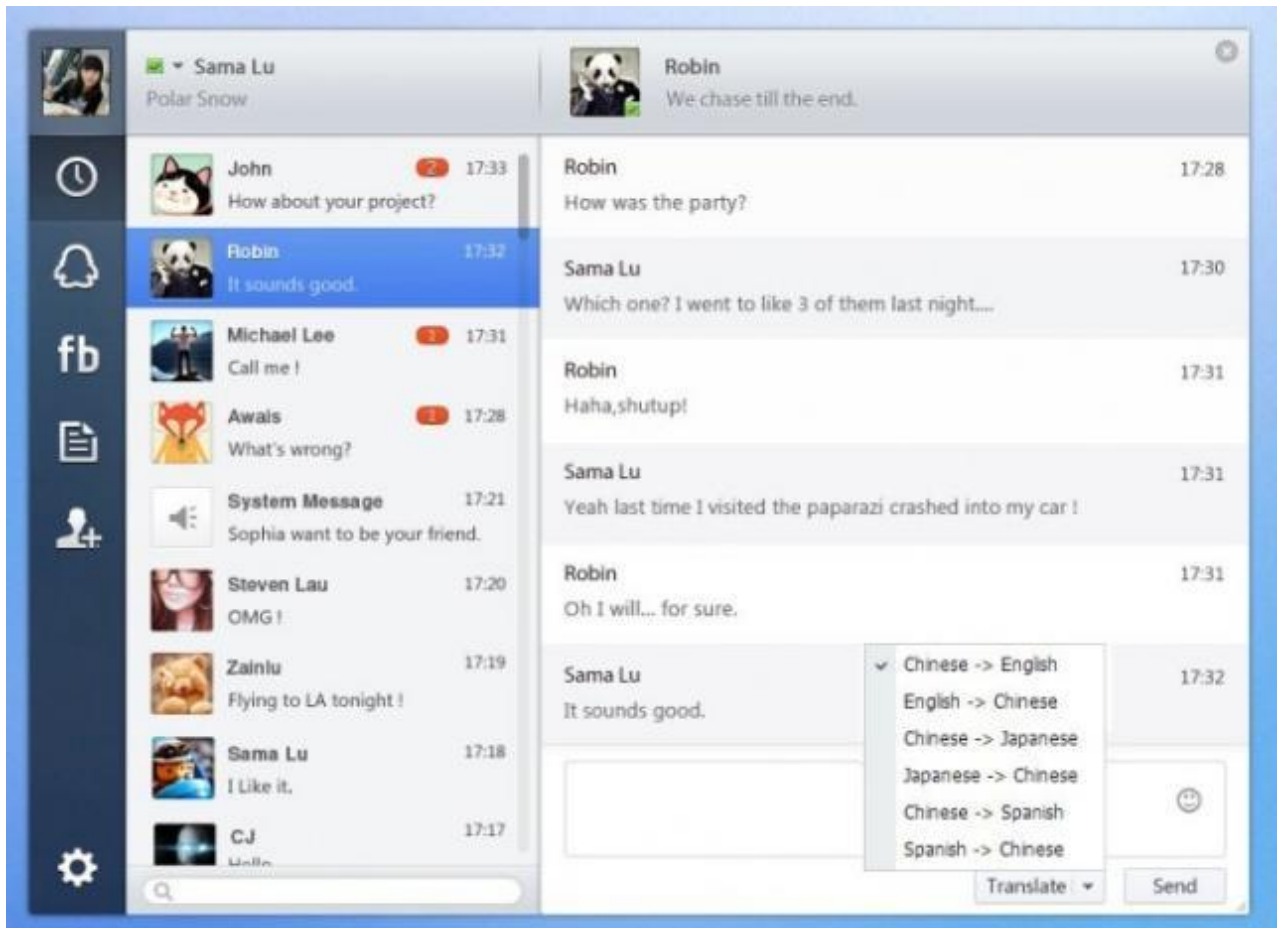


Рисунок 1.1 – Зображення інтерфейсу QQ

Аналіз аналога "WhatsUp"

"WhatsApp" - це одна з найпопулярніших месенджерських програм у світі. Вона була створена у 2009 році Брайаном Актоном та Яном Коумом, які колишні співробітники Yahoo. Пізніше компанія була придбана Facebook у 2014 році. (див. рисунок 1.2). Однак, додаток має складний інтерфейс, що може ускладнити його використання для менш досвідчених користувачів [7].



Рисунок 1.2 – Зображення інтерфейсу WhatsApp

Аналіз аналога "Facebook"

Facebook - це соціальна мережа, яка була створена Марком Цукербергом разом зі своїми однокурсниками у Гарвардському університеті у 2004 році [8]. Початково Facebook був призначений для студентів Гарварда, але згодом розширився на інші університети та остаточно став загальнодоступним в 2006 році. Однак, користувачі скаржаться на рекламу та застосунок має великі проблеми з безпекою користувацьких даних.

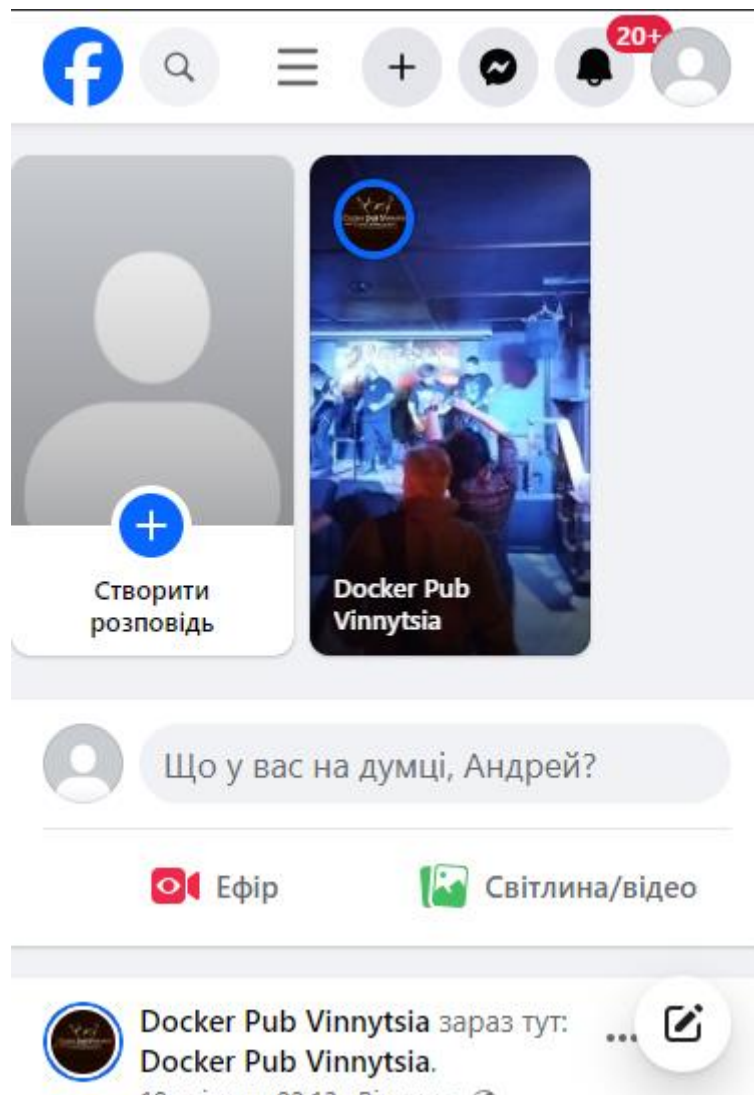


Рисунок 1.3 – Зображення інтерфейсу Facebook

Аналіз аналога "Myspace"

Myspace був однією з перших і найпопулярніших соціальних мереж, яка з'явилася ще до Facebook (див.рисунок 1.4). Його перевагою є широкий функціонал та зручний інтерфейс.

Проте, користувачі вказують на проблеми з безпекою даних та надмірну кількість реклами.

У свій час Myspace був дуже популярним серед молоді, музикантів, художників та інших творчих осіб [9]. Він став платформою для спілкування, відкриття нової музики та співпраці. Користувачі могли персоналізувати свої сторінки, додавати фотографії, відео, музику та інші мультимедійні матеріали.

Myspace поступово втратив популярність після запуску Facebook у 2004 році. Facebook пропонував кращий дизайн, більш простий інтерфейс, а також більшу привабливість для користувачів.

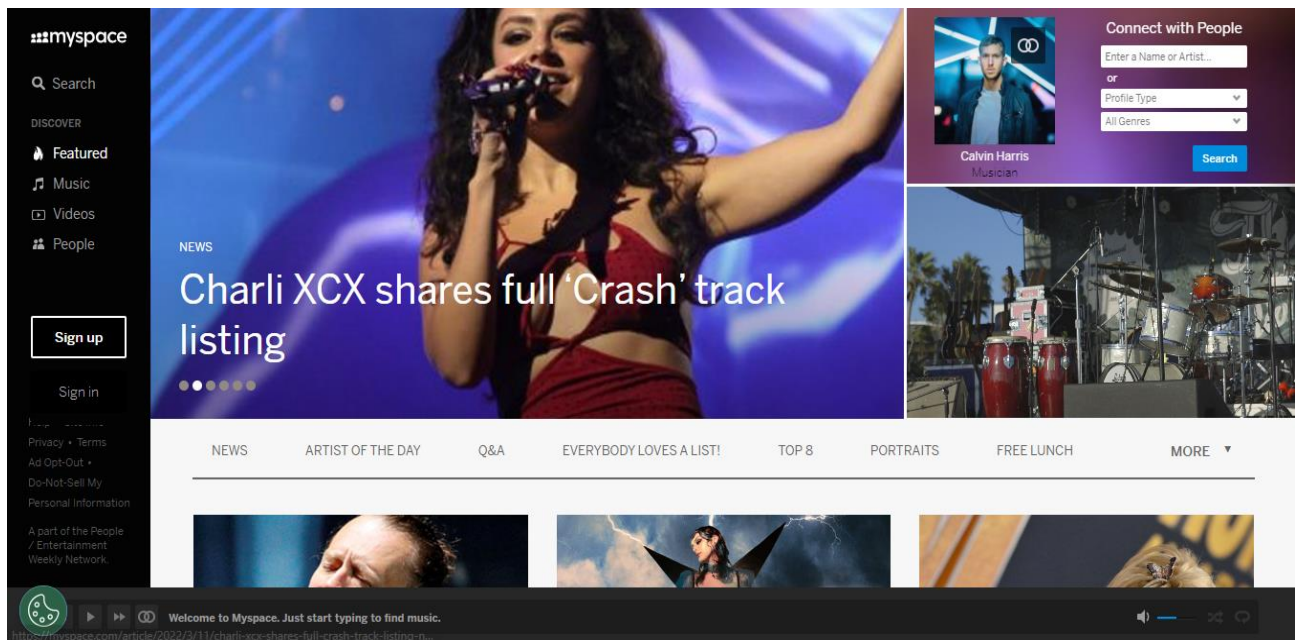


Рисунок 1.4 – Зображення інтерфейсу Myspace

Проведемо порівняльний аналіз цих чотирьох додатків, щоб сформулювати висновки:

QQ має застарілий інтерфейс, обмежений функціонал та погану безпеку даних, але має гарну стабільність роботи додатку.

WhatsApp обмежений функціоналом, має незручний інтерфейс, але є стабільним та безпечним додатком.

Facebook має розширений функціонал, простий інтерфейс та має стабільність роботи, але має проблеми з безпекою даних користувачів [10].

Myspace має надмірно перевантажений інтерфейс, проблеми зі стабільністю та безпекою даних своїх користувачів, але має розширений функціонал

На основі цього аналізу ми можемо створити таблицю, яка відображатиме всі переваги кожного додатку:

Таблиця 1.1 – Порівняльні характеристики додатків

Назва додатку	QQ	WhatsApp	Facebook	Myspase	akvavit
Інтеграція штучного інтелекту	0	0	0	0	1
Публічні публікації користувачів	1	1	1	1	1
Публічні повідомлення	0	0	1	1	1
Немає ліміту об'єму файлів	0	0	0	0	1
Сумарний коефіцієнт	1	1	2	2	4

Відповідно до таблиці порівняльних характеристик розробка власної інтерактивної системи передачі повідомлень є доцільною. Отриманий продукт надасть можливість розробникам та потенційним зацікавленим особам використовувати ефективну та надійну систему.

Така таблиця допоможе зробити обґрунтований вибір між різними додатками інтерактивних систем передачі повідомлень, а також надасть інформацію про новий додаток "akvavit".

Отже, розробка веб додатку інтерактивної системи передачі повідомлень відкриває широкі можливості для покращення взаємодії спільнот, створення унікального користувацького досвіду та підвищення рівня залученості та утримання аудиторії. Розуміючи потенційні можливості використання сучасних технологій та враховуючи вимоги до програмного забезпечення, компанії можуть розробляти інноваційні та привабливі вебсистеми, які використовують додаток для досягнення зростання та успішного функціонування.

1.3 Аналіз методів розв'язання задачі

Аналіз методів розв'язання поставленої задачі у контексті розробки веб додатку інтерактивної системи передачі повідомлень включає в себе ретельне вивчення різноманітних підходів та стратегій, які можна використовувати для досягнення успішного результату. Для цього аналізу потрібно пройти кілька етапів.

Починаючи з огляду існуючих методів, необхідно дослідити програмні продукти на ринку, спрямовані на задоволення потреб спілок. Це включає вивчення їх функціональності, інтерфейсу та рівня задоволення користувачів. Крім того, важливо провести аналіз літературних джерел, статей та досліджень, що стосуються розробки веб додатків для цієї категорії користувачів.

На другому етапі потрібно провести дослідження потреб та вимог потенційних користувачів. Це означає вивчення типових сценаріїв використання, визначення основних функціональних потреб користувачів та їх очікувань від додатку. Для цього можуть здійснюватися спеціальні дослідження та опитування серед спілок.

Наступним кроком є вибір відповідних технологій для розробки додатку. Це включає вибір платформи розробки, мов програмування, бази даних та інших технічних аспектів.

Далі необхідно розробити архітектуру веб додатку, визначивши структуру, модулі та компоненти. Особлива увага повинна бути приділена розробці зручного та ефективного інтерфейсу користувача, який відповідає потребам цільової аудиторії.

Після проєктування архітектури настав час для розробки самого додатку, включаючи програмування функціоналу, інтеграцію штучного інтелекту та розробку інтерфейсу. Після завершення розробки необхідно провести тестування для перевірки правильності та стабільності роботи додатку.

Останнім етапом є впровадження веб додатку серед спільнот та підтримка користувачів, що включає вирішення можливих проблем, оновлення програмного забезпечення та відповідь на запитання користувачів.

Розглянувши переваги та недоліки кожного методу, було вирішено розробити веб додаток інтерактивної системи передачі повідомлень, який враховуватиме основні вимоги та потреби цільової аудиторії.

1.4 Постановка задач проєкту

Після уважного аналізу переваг та недоліків існуючих систем передачі повідомлень було сформульовано наступні завдання, які необхідно виконати для успішної розробки додатку:

1. провести дослідження та аналіз потреб та очікувань спільнот щодо веб додатку;
2. визначити основні функціональні потреби та встановити пріоритети користувачів;
3. обрати платформу розробки;
4. розробити структуру додатку, включаючи його модулі та компоненти, для забезпечення оптимального функціоналу;
5. написати код для реалізації функціоналу додатку;
6. інтегрувати штучний інтелект
7. розробити інтерфейс користувача;
8. провести комплексне тестування додатку на різних браузерях та в різних умовах для забезпечення стабільності та ефективності його роботи;

1.5 Висновки

У першому розділі детально розглянуто різні аспекти розробки веб застосунку інтерактивної системи передачі повідомлень. Почавши з аналізу

потреб та очікувань користувачів, виявились основні функціональні потреби, які повинен вирішувати додаток. Дослідження ринку дозволило виявити наявні системи та їх переваги та недоліки, що видалося надзвичайно корисним у формулюванні власних стратегій.

Основний акцент був зроблений на виборі оптимальної платформи розробки, і вирішено зосередитися на веб додатку, зважаючи на широке поширення веб додатків серед цільової аудиторії. Крім того, визначено ключові етапи розробки додатку, включаючи розробку його структури, написання коду, розробку інтерфейсу користувача та проведення тестування.

Загалом, перший розділ надав чітке уявлення про напрямки подальших дій у процесі розробки веб додатку інтерактивної системи передачі повідомлень.

2 РОЗРОБКА ІНТЕРФЕЙСУ НА МОДЕЛЕЙ СИСТЕМИ

2.1 Аналіз вхідних даних системи

Аналіз вхідних даних системи є ключовим етапом у розробці інтерактивної системи передачі повідомлень. Цей процес полягає у ретельному розгляді та розумінні різних типів інформації, яка може надходити до системи через користувачів або з інших джерел.

Особиста інформація користувача є основним типом вхідних даних, яку варто розглянути. Ця інформація включає ім'я та прізвище, електрону пошту, номер телефону, дату народження, та інформацію про себе. Ці дані не лише допомагають ідентифікувати користувача, та й створюють інтерес серед інших користувачів для спілкування з ним.

Також потрібно враховувати дані про заявки та скарги користувачів, які надходять через веб додаток. Ця інформація є дуже цінною, оскільки дозволяє оперативно виявляти проблеми та недоліки в додатку. Ретельний аналіз цих даних дозволяє не лише вчасно реагувати на проблеми, але й вдосконалювати додаток та підвищувати задоволення користувачів.

Узагальнюючи, аналіз вхідних даних допомагає зрозуміти потреби користувачів, оптимізувати використання ресурсів та покращувати якість. Це важливий етап у процесі розробки веб додатку, оскільки від правильного аналізу цих даних залежить успішність та ефективність роботи програмного забезпечення.

2.2 Розробка інтерфейсу програмного додатку

Розробка інтерфейсу додатку - це важливий етап у процесі створення інтерактивної системи передачі повідомлень. Інтерфейс визначає спосіб, яким користувачі взаємодіють з програмою, тому він повинен бути зручним, естетично привабливим та функціональним [11].

Структурна схема є графічною моделлю, яка демонструє складові частини системи та взаємозв'язки між ними. Вона служить для візуалізації структури системи, дозволяючи краще зрозуміти її компоненти і їхні взаємодії. Структурна схема є важливим інструментом для аналізу, проектування і обслуговування систем, оскільки вона надає наочне уявлення про її будову та функціонування.

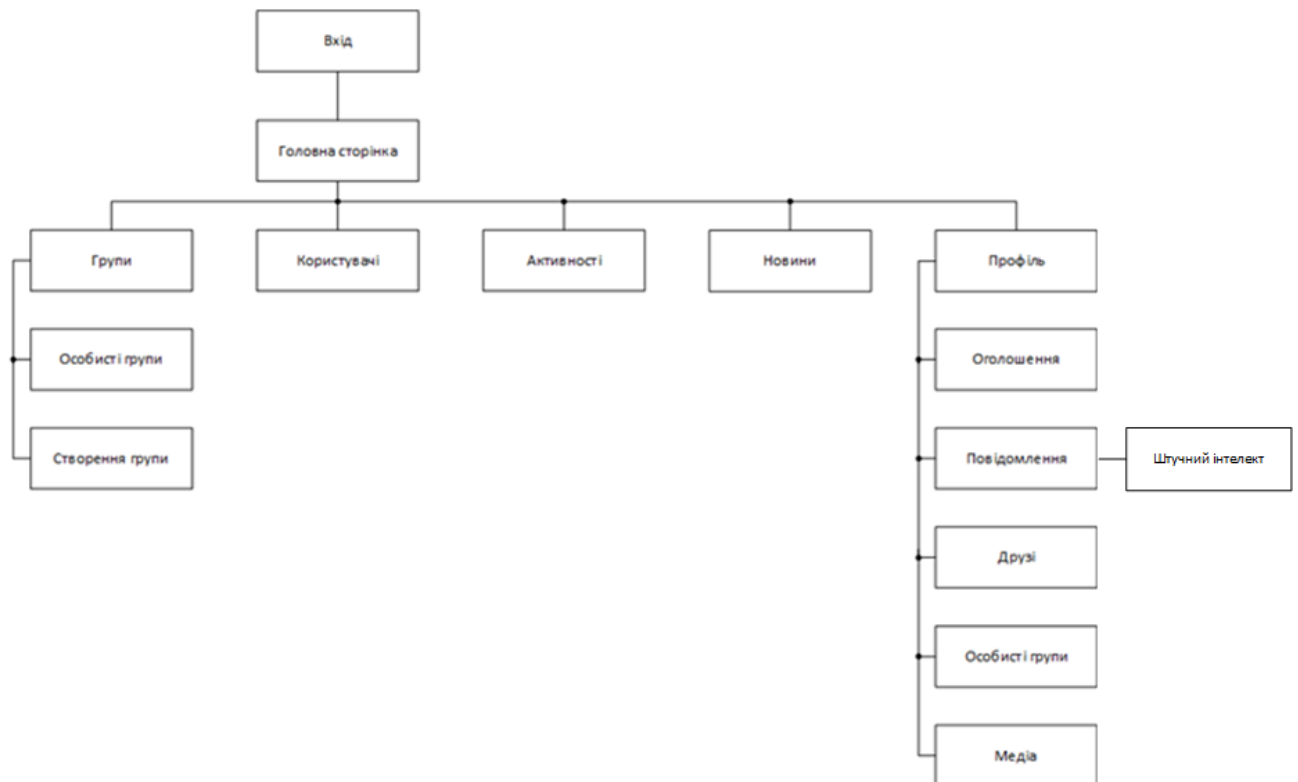


Рисунок 2.1 – Структурна схема системи

Сторінка входу користувача до системи складається із двох полів введення даних, кнопки входу, кнопки “Запам’ятати мене”, кнопки відновлення паролю та кнопки реєстрації. На рисунку 2.2 показано графічну схему сторінки входу, а на рисунку 2.3 розроблений інтерфейс.

The diagram shows a login page layout within a rectangular frame. The elements are numbered as follows:

- 1: The top-left corner of the page frame.
- 2: A horizontal input field for the username.
- 3: A horizontal input field for the password.
- 4: A small square checkbox.
- 5: A horizontal button for login.
- 6: A horizontal button for registration.
- 7: A horizontal button for password recovery.

Рисунок 2.2 – Графічна схема вебсторінки авторизації до системи

Елементи сторінки входу до системи:

1. Робоча область.
2. Поле вводу логіну.
3. Поле вводу пароля.
4. Кнопка “Запам’ятати мене”.
5. Кнопка входу
6. Кнопка реєстрації.
7. Кнопка відновлення паролю

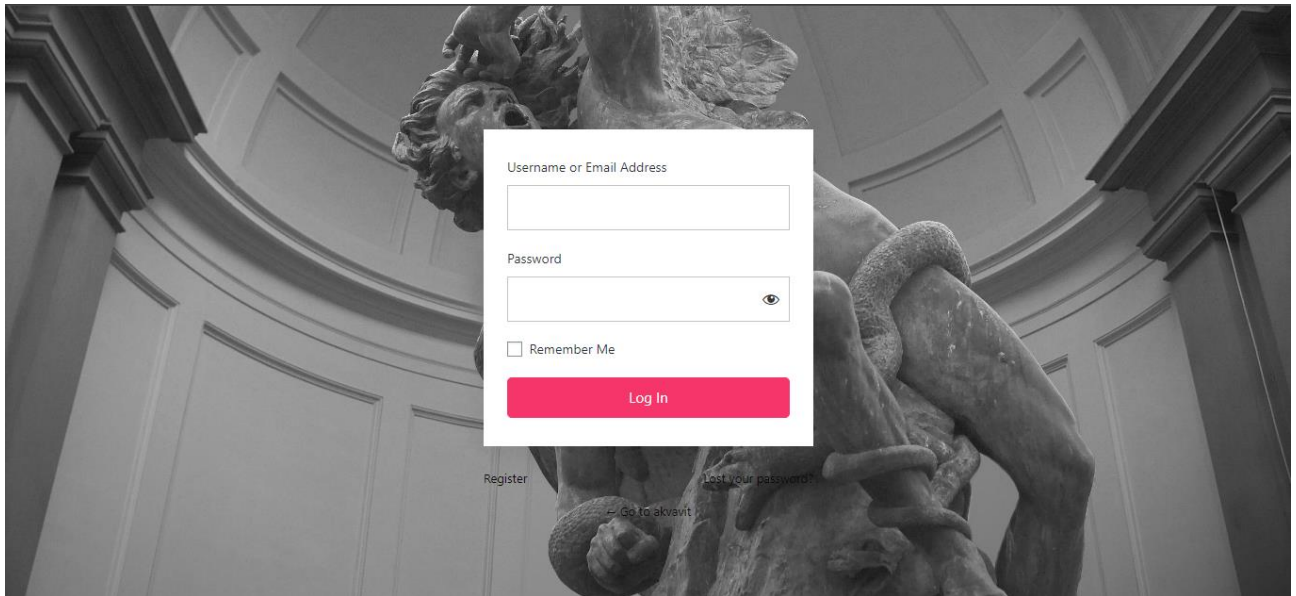


Рисунок 2.3 – Вебсторінка авторизації до системи

Для усіх сторінок вебсайту, окрім, сторінки входу та реєстрації було розроблено універсальний верхній блок, який в собі вмістив логотип, кнопку головної сторінки, кнопку груп, кнопку користувачі, кнопку новини, кнопку пошуку, кнопку повідомлень, кнопку оголошень та кнопку профілю. На рисунку 2.4 показано графічну схему верхнього блоку, а на рисунку 2.5 розроблений інтерфейс.

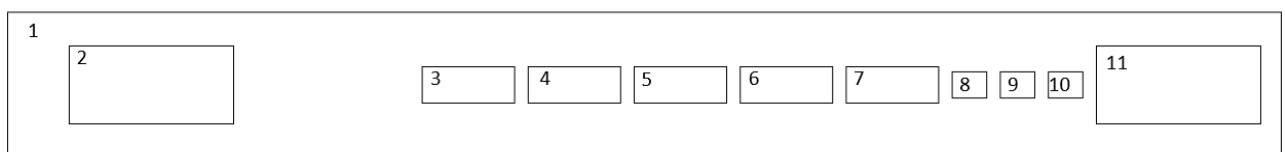


Рисунок 2.2 – Графічна схема глобального заголовку вінка

Елементи верхнього блоку:

1. Робоча область.
2. Лого.
3. Кнопка Головна сторінка.
4. Кнопка Групи.
5. Кнопка Користувачі.

6. Кнопка Активностей.
7. Кнопка Новини.
8. Кнопка Пошук.
9. Кнопка Повідомлення.
10. Кнопка Оголошення.
11. Кнопка Профіль.

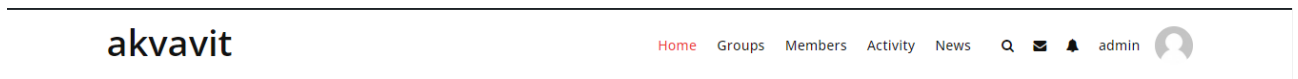


Рисунок 2.5 – Глобальний заголовок вікна

Головна сторінка є ключовим елементом у формуванні ідентичності та визначенні враження від використання додатку [12]. Ця сторінка показує потенційному користувачу спільноту системи. На рисунку 2.6 показано графічну схему головної сторінки, а на рисунку 2.7 розроблений інтерфейс.

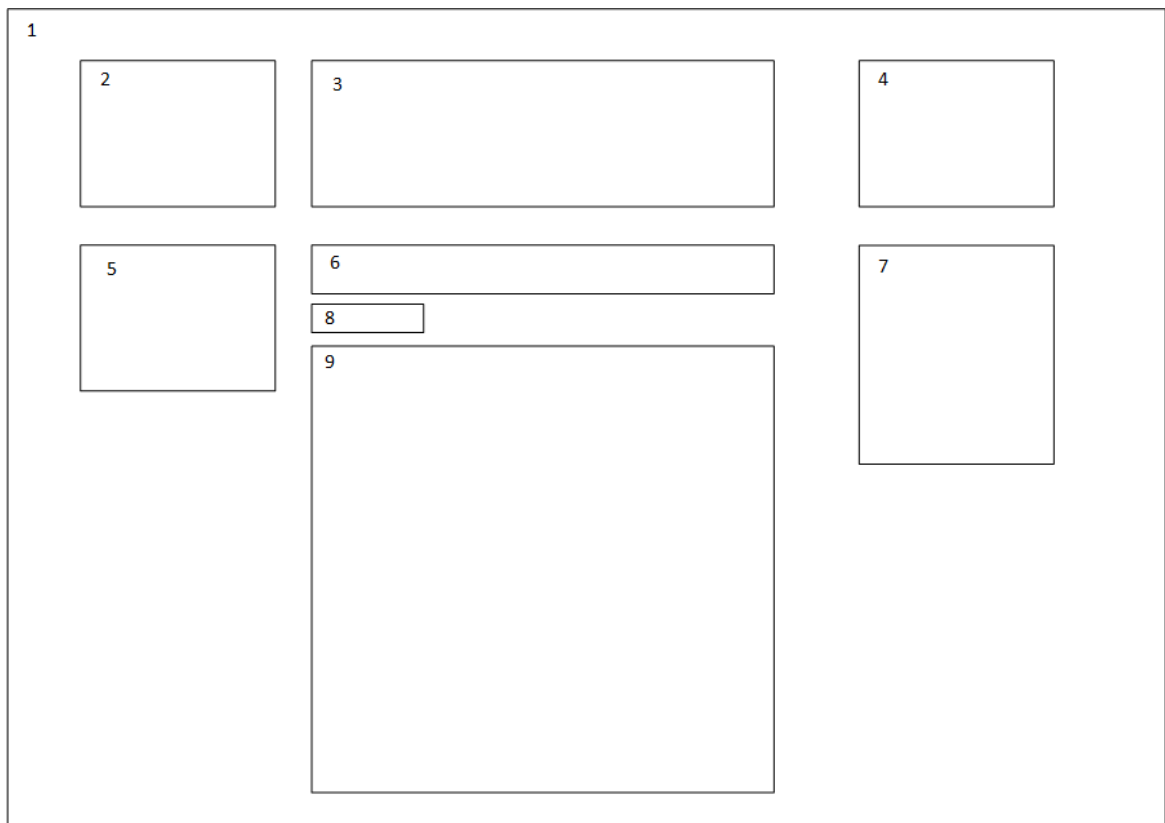


Рисунок 2.6 – Графічна схема головної вебсторінки

Елементи головної сторінки:

1. Робоча область.
2. Вікно груп.
3. Вікно створення посту.
4. Вікно заповнення профілю.
5. Вікно недавньо активних користувачі.
6. Фільтри пошуку.
7. Вікно списку користувачі.
8. Вікно пошуку.
9. Область публікацій.

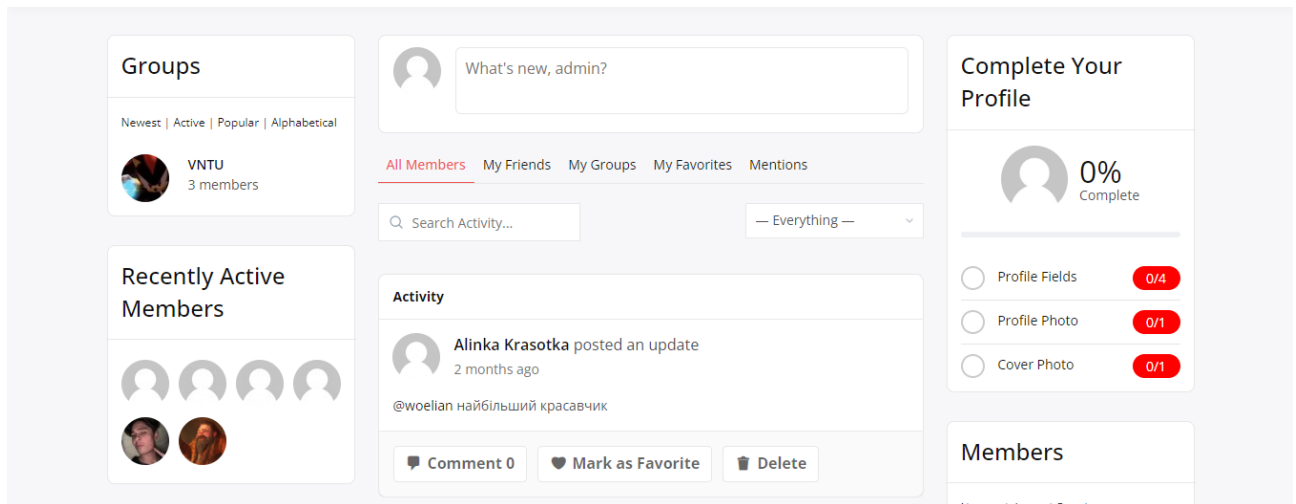


Рисунок 2.7 – Головна вебсторінка

Особисті повідомлення є ключовою функцією вебсайту. Вікно для таких повідомлень слід розробити таким чином, щоб воно забезпечувало зручний та інтуїтивно зрозумілий інтерфейс для користувачів. Необхідно передбачити можливість швидкого доступу до листування, підтримку різних форматів повідомлень (текст, зображення, файли), а також функціональність для групових чатів та миттєвих повідомлень. Інтерфейс має бути адаптивним, щоб однаково добре працювати як на десктопних, так і на мобільних пристроях. На рисунку 2.8 показано графічну схему особистих повідомлень, а на рисунку 2.9 розроблений інтерфейс.

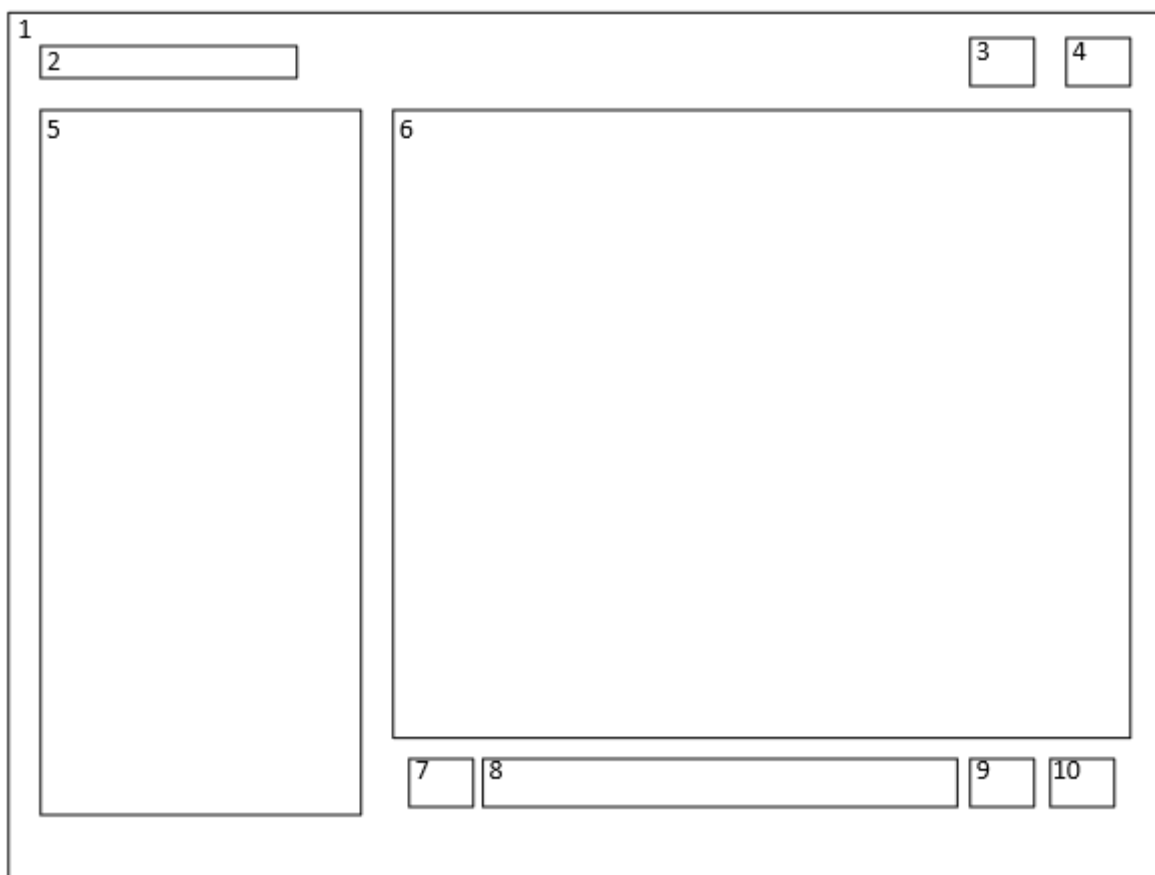


Рисунок 2.8 – Графічна схема особистих повідомлень

Елементи головної сторінки:

1. Робоча область.
2. Вікно пошуку.
3. Кнопка збережених повідомлень.
4. Кнопка На весь екран.
5. Список чатів.
6. Історія повідомлень.
7. Кнопка прикріплення файлу.
8. Область друку повідомлення.
9. Кнопка емодзі.
10. Кнопка надсилання повідомлення.

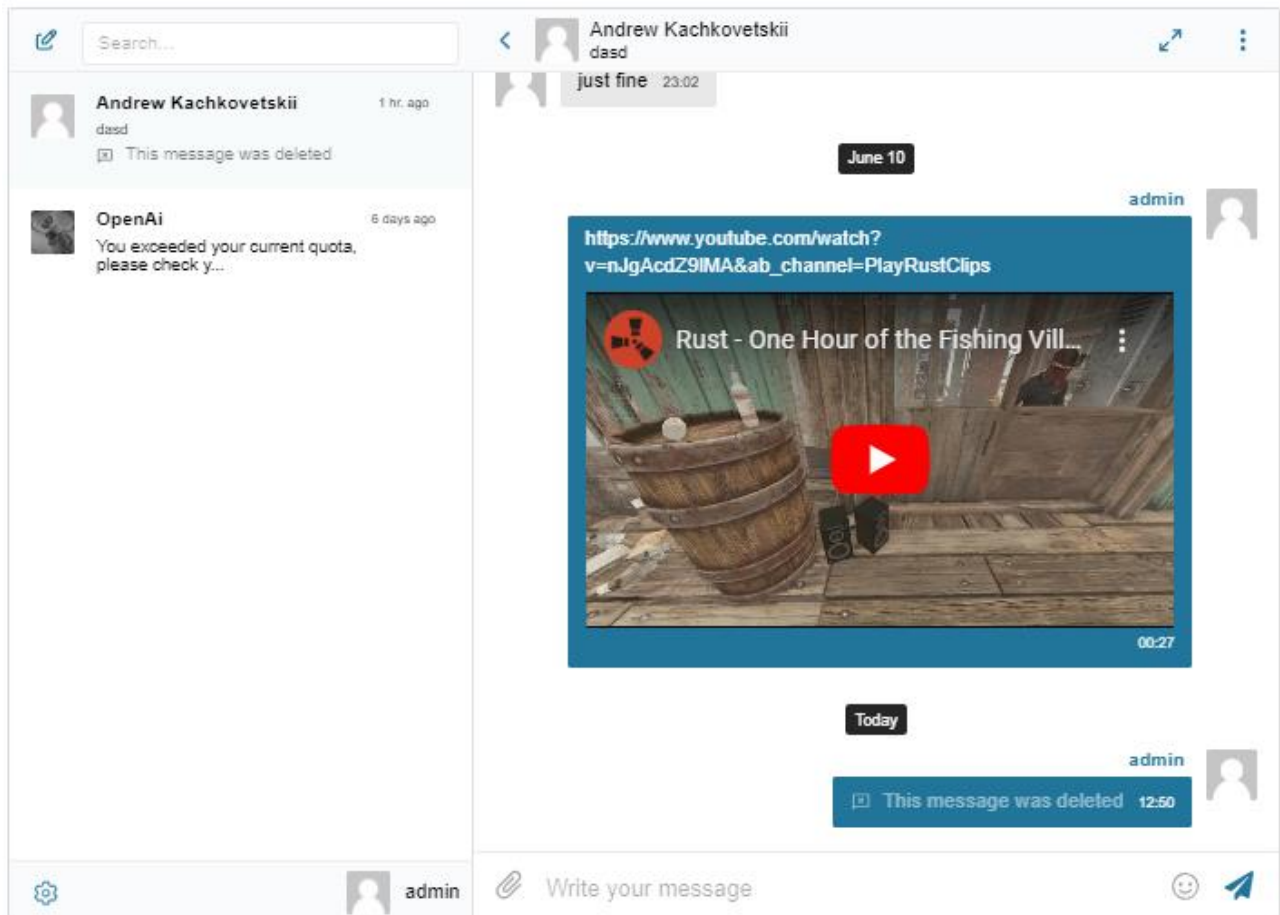


Рисунок 2.9 – Особисті повідомлення

2.3 Розробка моделі системи

Для кращого уявлення про роботу програмного додатку ми вирішили підготувати UML-діаграму діяльності, що зображує послідовність операцій в системі (див. рисунок 2.17). Спочатку користувач авторизується у додатку [13].

Після успішної авторизації він переходить на головний екран додатку, де може переглянути загальну інформацію або доступні функції. З цього екрану він може перейти до різних розділів додатку, таких як групи, список користувачів, новин, повідомлень та в свій профіль, натискаючи на відповідні елементи інтерфейсу.

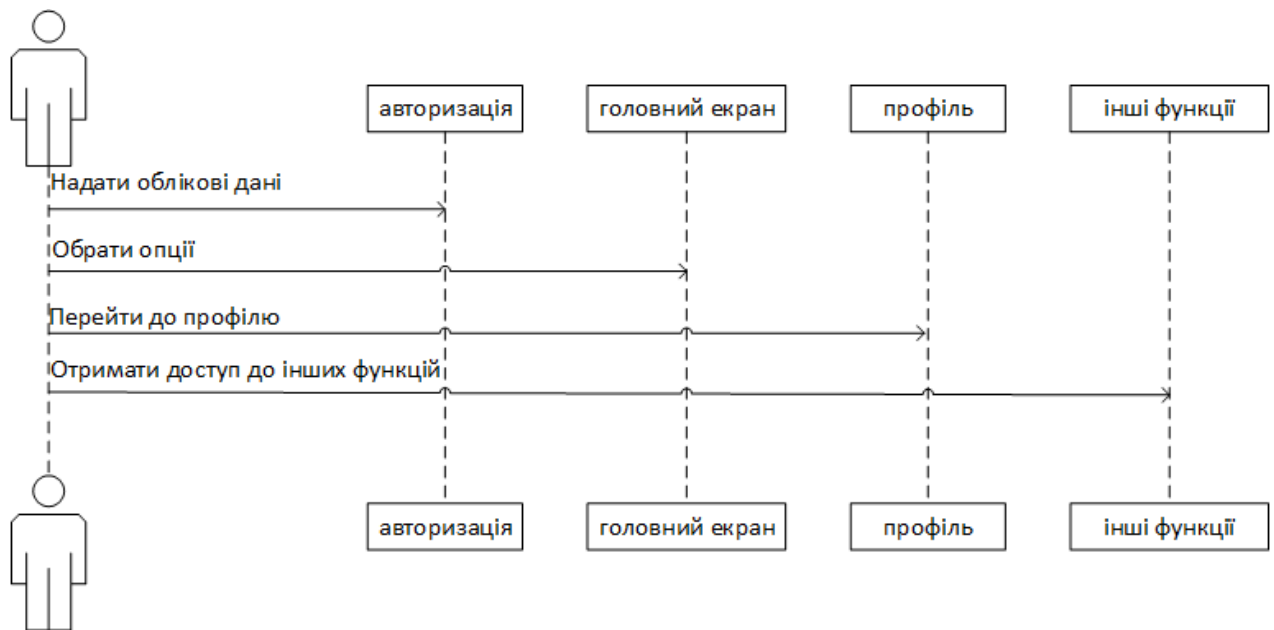


Рисунок 2.17 – Діаграма діяльності системи

2.4 Розробка модулів для роботи системи

Для розробки веб додатку необхідно розробити загальні модулі, які визначатимуть послідовність дій користувача в додатку (див. рисунок 2.18). Згідно цього модулю, спочатку користувач пройде авторизацію, після чого потрапить на головний екран. Далі, він матиме можливість перейти до груп, списку користувачів, новин, повідомлень та в свій профіль [14]. Таким чином, після входу в систему, запускається модуль перегляду головної сторінки та доступ до функцій.

Якщо користувач перейде до свого профілю, він матиме можливість переглянути свої дані, повідомлення, групи, медіафайли, а також редагувати всі свої дані, окрім логіну. Це означає, що він може оновити інформацію про себе, змінити фотографію профілю, налаштувати приватність, а також переглянути історію активності або редагувати свій статус. Такий функціонал дозволяє користувачам зберігати та керувати своїми особистими даними зручно та ефективно.

У випадку, якщо користувач залишиться на головному екрані, він матиме можливість переглянути розділи групи, користувачі, новини. Крім того, на

головному екрані можуть бути доступні такі корисні функції від стрічки публікацій до користувачів які знаходяться онлайн. Такий підхід дозволяє користувачу швидко знаходити необхідну інформацію та зручно користуватися додатком [15].

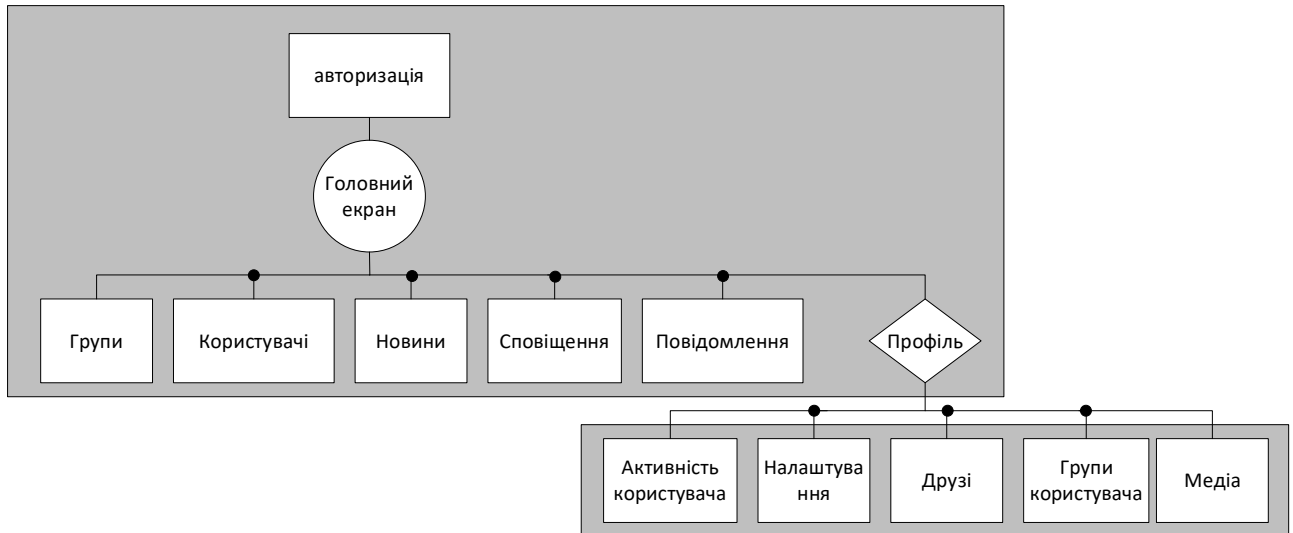


Рисунок 2.18 – Діаграма послідовності

2.5 Висновки

У другому розділі проведено аналіз вхідних та вихідних даних програмного застосунку для визначення необхідного функціоналу та потреб користувачів. Алгоритм роботи вебзастосунку розроблено з урахуванням потреб користувачів та особливостей функціоналу.

Розроблено модель роботи програмного застосунку за допомогою UML діаграм для візуалізації взаємозв'язків між компонентами системи та процесами. Це дозволило краще зрозуміти структуру та логіку роботи програми.

3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ В ВЕБДОДАТКУ

3.1 Варіантний аналіз і обґрунтування вибору методів та засобів для реалізації програмного забезпечення

JavaScript є однією з найбільш використовуваних мов програмування, особливо для фронтенд-розробки [16]. Його головна перевага полягає в тому, що він виконується на стороні клієнта, що дозволяє створювати динамічні та інтерактивні інтерфейси. Крім того, з появою Node.js, JavaScript став популярним і на стороні сервера, що забезпечує повноцінну розробку на одній мові. Однак, JavaScript має свої обмеження, такі як складність в підтримці великих проєктів, асинхронність, яка може бути важкою для розуміння новачками, та неоднорідність екосистеми, що іноді призводить до плутанини в інструментах та фреймворках.

Python - це мова програмування, яка стрімко набирає популярність завдяки своїй простоті та читабельності коду. Вона широко використовується в різних галузях, таких як наукові дослідження, аналіз даних, машинне навчання, штучний інтелект та веб-розробка. Django і Flask - це два популярні фреймворки для веб-розробки на Python, які дозволяють швидко створювати надійні та масштабовані веб-додатки. Django забезпечує "все включено" підхід з багатьма вбудованими інструментами, тоді як Flask є мікрофреймворком, що надає більше свободи у виборі компонентів. Однак, продуктивність Python може бути нижчою порівняно з іншими мовами, і він не завжди є оптимальним вибором для високонавантажених систем. Також слід зазначити, що Python інколи може вимагати більше ресурсів для виконання завдань, що може стати проблемою для деяких проєктів.

Ruby є відомим завдяки своєму фреймворку Ruby on Rails, який спрощує розробку веб-додатків і пропонує високу продуктивність. Ruby відомий своєю елегантністю та простотою синтаксису, що робить його привабливим для багатьох розробників. Rails пропонує набір інструментів, що полегшують

створення CRUD-додатків (створення, читання, оновлення, видалення), а також підтримку RESTful архітектури, що робить його дуже потужним інструментом для веб-розробки. Тим не менш, Ruby може мати проблеми з продуктивністю і масштабованістю, що може обмежувати його використання в дуже великих проєктах. Деякі розробники також зазначають, що Ruby on Rails може бути занадто "магічним", приховуючи важливі деталі реалізації, що може ускладнити налагодження та оптимізацію коду.

PHP є одним з найстаріших та найпопулярніших мов програмування для веб-розробки [17]. Він був спеціально створений для розробки веб-додатків і добре інтегрується з HTML, що дозволяє легко вставляти PHP-код безпосередньо в HTML-документи. PHP має величезну кількість бібліотек та фреймворків, таких як Laravel, Symfony, CodeIgniter та Yii, що дозволяють швидко розробляти складні веб-додатки. Однією з головних переваг PHP є його простота у вивченні та використанні, що робить його доступним для новачків. Велика спільнота розробників надає підтримку, ресурси та численні готові рішення для типових завдань, що значно спрощує процес розробки. Крім того, PHP відомий своєю сумісністю з багатьма базами даних, такими як MySQL, PostgreSQL, SQLite та іншими, а також підтримкою різних веб-серверів, включаючи Apache та Nginx, що робить його дуже гнучким інструментом для веб-розробки.

PHP також має ряд інших переваг, які роблять його фаворитом серед мов програмування для веб-розробки. Наприклад, PHP широко використовується у великих веб-проєктах, таких як Facebook і Wikipedia, що свідчить про його здатність обробляти високі навантаження та забезпечувати надійну роботу. Крім того, PHP має потужні засоби для роботи з файлами, сесіями, куки та іншими аспектами веб-розробки, що дозволяє створювати повнофункціональні веб-додатки з мінімальними зусиллями. Загалом, PHP залишається фаворитом для розробки веб-додатків завдяки своїй простоті, гнучкості та широкій підтримці. Хоча інші мови мають свої переваги в різних контекстах, PHP пропонує збалансоване поєднання продуктивності, простоти і надійності, що робить його чудовим вибором для багатьох проєктів. Його здатність легко інтегруватися з

різними технологіями, широке ком'юніті та велика кількість готових рішень роблять PHP універсальним інструментом, який підходить для різних видів веб-розробки, від невеликих блогів до великих корпоративних додатків.

Порівняльний аналіз описаних мов програмування наведено в таблиці 3.1

Таблиця 3.1 – Порівняння мов програмування

	PHP	Ruby	Python	JavaScript
Продуктивність	1	0	0	1
Синтаксис	1	1	1	0
Масштабованість	1	0	1	0
Документація та підтримка	1	1	1	1
Використання веб-фреймворків	1	1	1	1
Робота з файловою системою	1	1	1	1
Асинхронні можливості	1	1	1	1
Інтеграція з іншими сервісами	1	1	1	1
Сумарний коефіцієнт	8	6	7	6

WordPress є найбільш популярною CMS у світі, на яку припадає понад 40% всіх веб-сайтів [18]. Його успіх зумовлений кількома ключовими факторами. По-перше, WordPress має надзвичайно зручний інтерфейс, що робить його доступним для новачків. Навіть користувачі без технічного досвіду можуть легко

створювати та управляти веб-сайтами завдяки інтуїтивно зрозумілій панелі адміністрування. По-друге, WordPress пропонує велику кількість безкоштовних та платних тем і плагінів, що дозволяє розширювати функціональність сайту без необхідності програмування. Це робить його надзвичайно гнучким та адаптивним до різних потреб.

Joomla є другою за популярністю CMS, яка надає більше можливостей для налаштування та складніших функцій, порівняно з WordPress. Joomla підходить для створення більш складних сайтів з багаторівневою структурою. Вона також має потужну систему управління користувачами та підтримує різні види контенту. Однак, для новачків Joomla може бути складнішою у використанні, а її налаштування можуть вимагати більше технічних знань.

На рисунку 3.1 зображено інтерфейс системи керування змістом Joomla.

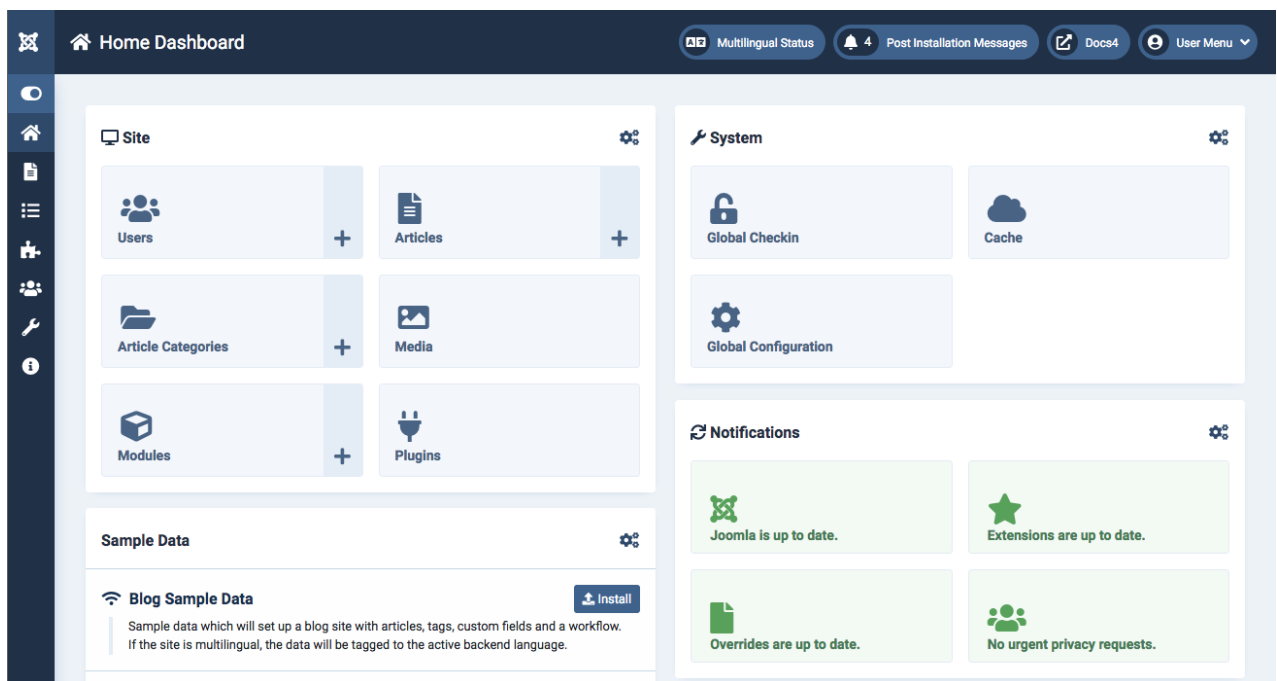


Рисунок 3.1 – Інтерфейс Joomla

Drupal - це потужна CMS, що найкраще підходить для великих та складних проектів з високими вимогами до безпеки та продуктивності. Drupal має дуже гнучку систему таксономії, яка дозволяє створювати складні структури контенту. Вона також відома своєю надійною безпекою, що робить її

популярною серед урядових та корпоративних сайтів. Однак, Drupal має круту криву навчання і вимагає значних технічних знань для налаштування та адміністрування.

На рисунку 3.2 зображено інтерфейс системи керування змістом Drupal.

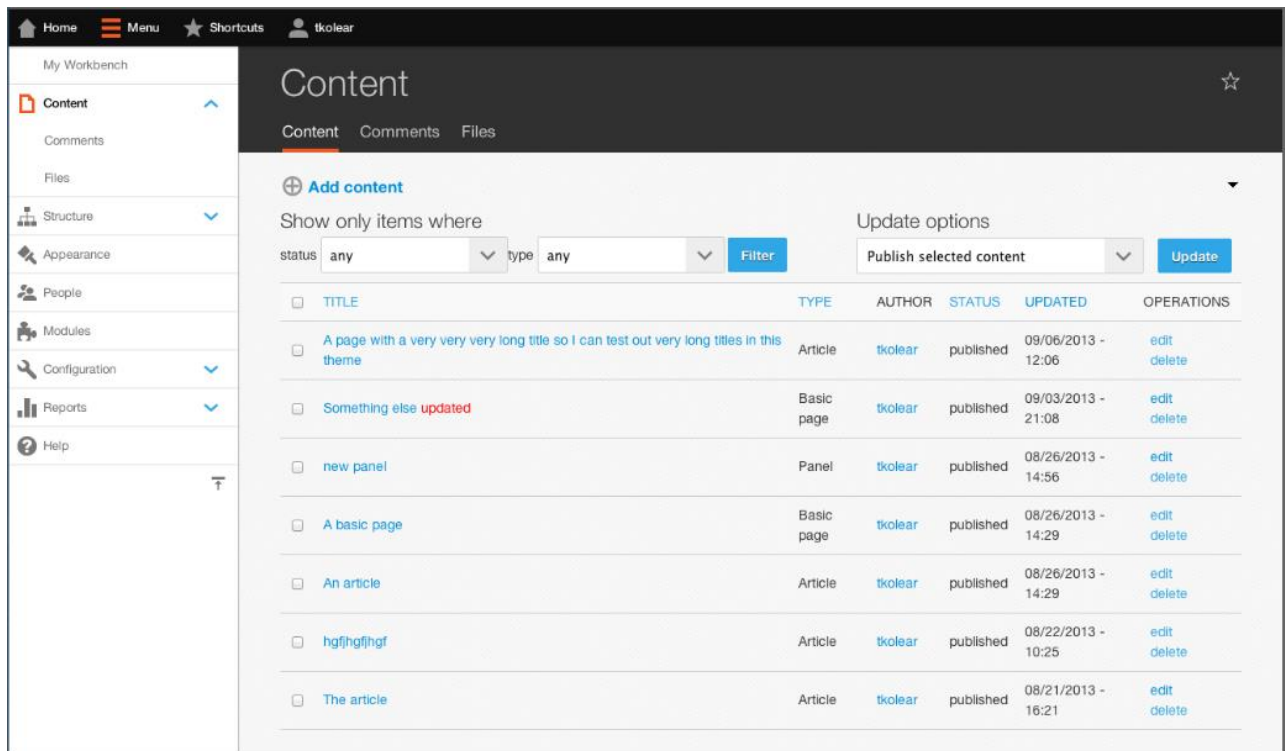


Рисунок 3.2 – Інтерфейс Drupal

Magento є спеціалізованою CMS для електронної комерції. Вона пропонує широкий спектр функцій для управління інтернет-магазинами, включаючи управління продуктами, замовленнями, платежами та доставкою. Magento ідеально підходить для великих інтернет-магазинів з великим обсягом товарів і складною логістикою. Однак, її складність і вимоги до ресурсів можуть бути занадто високими для малих та середніх підприємств.

На рисунку 3.3 зображено інтерфейс системи керування змістом Magento.

Products

Add Product

Filters Default View Columns

Active filters: Attribute Set: Gear

Search by keyword

Actions 20 records found 20 per page 1 of 1

	ID	Thumbnail	Name	Type	Attribute Set	SKU	Price	Quantity	Visibility	Status	Website
☐	37		Endurance Watch	Simple Product	Gear	24-MG01	\$49.00	100.0000	Catalog, Search	Enabled	Main Website
☐	40		Dash Digital Watch	Simple Product	Gear	24-MG02	\$92.00	100.0000	Catalog, Search	Enabled	Main Website
☐	38		Summit Watch	Simple Product	Gear	24-MG03	\$54.00	100.0000	Catalog, Search	Enabled	Main Website
☐	36		Aim Analog Watch	Simple Product	Gear	24-MG04	\$45.00	100.0000	Catalog, Search	Enabled	Main Website

Рисунок 3.3 – Інтерфейс Magento

WordPress виділяється серед цих CMS завдяки своїй простоті використання, великій кількості тем і плагінів, а також активній спільноті користувачів та розробників. Він підходить для широкого спектру проєктів, від блогів і персональних сайтів до корпоративних порталів і інтернет-магазинів. WordPress постійно оновлюється і вдосконалюється, що забезпечує його актуальність та безпеку. Крім того, він має потужну SEO-оптимізацію з коробки та інтеграцію з різними соціальними мережами і маркетинговими інструментами. У підсумку, хоча кожна з розглянутих CMS має свої унікальні переваги і підходить для певних типів проєктів.

На рисунку 3.4 зображено інтерфейс системи керування змістом WordPress.

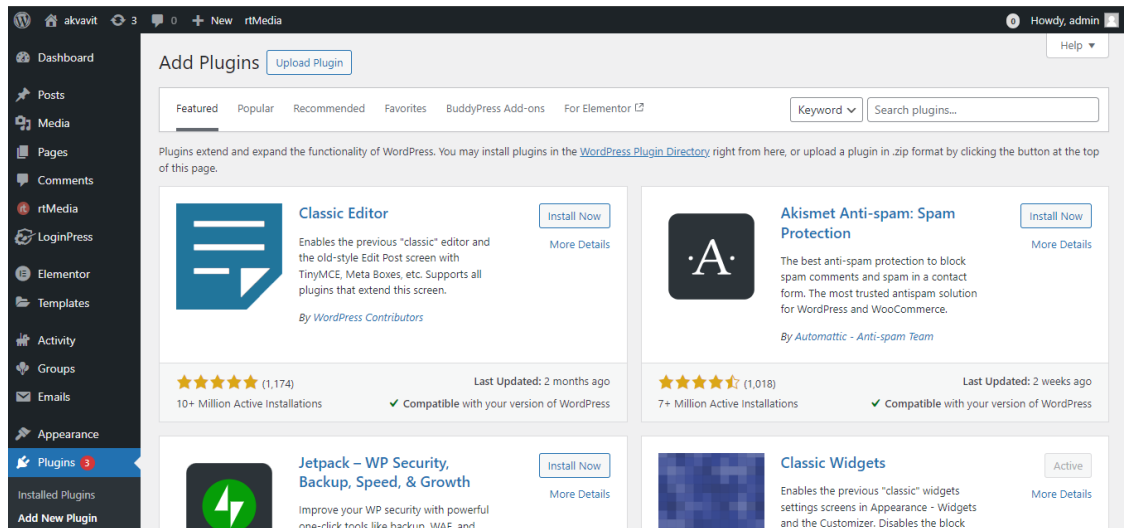


Рисунок 3.4 – Інтерфейс WordPress

WordPress є фаворитом завдяки своїй універсальності, зручності використання та широкій підтримці спільноти. Це робить його ідеальним вибором для більшості веб-проектів, особливо для тих, хто хоче швидко і без значних витрат створити функціональний і привабливий веб-сайт.

Порівняльний аналіз описаних систем керування змісту наведено в таблиці 3.2

Таблиця 3.2 – Порівняння систем керування змісту

	WordPress	Magento	Drupal	Joomla
Зручність використання	1	0	0	0
Швидкодія	1	0	0	0
Спільнота та підтримка	1	1	1	1
Безпека	1	0	1	1
Можливості багатомовності	1	0	1	1
Можливості SEO	1	0	0	0
Інтеграція з соціальними мережами	1	0	0	0
Легкість кастомізації	1	1	1	1
Сумарний коефіцієнт	8	2	4	4

Отже, вибір технологій для розробки інтерактивної системи передачі повідомлень, що використовує інструменти PHP і WordPress, має вирішальне значення для створення надійної інтерактивної системи передачі повідомлень. Поєднання цих технологій надає всі необхідні інструменти для розробки ефективної, надійного та простого в обслуговуванні веб додатку.

3.2 Розробка головного екрану та навігації

Розробка головного екрану та навігації в веб додатку інтерактивної системи передачі повідомлень відіграє ключову роль у забезпеченні зручного та ефективного користувацького досвіду. Головний екран є центральним взаємодійовим майданчиком, через який користувачі мають доступ до основних функцій системи та стрічки активностей спільноти.

На головному екрані веб додатку розміщені основні елементи керування та інформаційні блоки, які найбільш важливі для користувачів. Кожен елемент на головному екрані чітко позначений та доступний для використання з мінімальною кількістю клацань.

Щодо навігації, вона повинна бути легко зрозумілою та інтуїтивно зрозумілою для користувачів. Меню навігації може включати в себе розділи, такі як 'Групи', 'Користувачі', 'Активність', 'Новини', 'Повідомлення', 'Профіль'. Користувачі повинні мати можливість легко переходити між цими розділами та знаходити необхідну інформацію.

Забезпечення зручності та ефективності користувацького досвіду на головному екрані та під час навігації вимагає ретельного проєктування та тестування. Це дозволяє створити додаток, який задовольняє потреби користувачів системи передачі повідомлень та надає їм зручний інструмент для взаємодії [19]. На рисунку 3.5 зображено ініціалізацію головного екрану.

```

doctype html>
html lang="en-US" class="no-js">
head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1">
  <link rel="profile" href="https://gmpg.org/xfn/11">

  <title>Activity &#8211; akvavit</title>
  <meta name='robots' content='max-image-preview:large' />
  <link rel='dns-prefetch' href='//fonts.googleapis.com' />
  <link href='https://fonts.gstatic.com' crossorigin rel='preconnect' />
  <link rel="alternate" type="application/rss+xml" title="akvavit &raquo; Feed"
    href="https://andrewcool.pp.ua/feed/" />
  <link rel="alternate" type="application/rss+xml" title="akvavit &raquo; Comments Feed"
    href="https://andrewcool.pp.ua/comments/feed/" />
  <style id='wp-emoji-styles-inline-css' type='text/css'>
    img.wp-smiley,
    img.emoji {
      display: inline !important;
      border: none !important;
      box-shadow: none !important;
      height: 1em !important;
      width: 1em !important;
      margin: 0 .07em !important;
      vertical-align: -.1em !important;
      background: none !important;
      padding: 0 !important
    }
  </style>
  <link rel='stylesheet' id='wp-block-library-css'

```

Рисунок 3.5 – Ініціалізацію головного екрану

Після переходу на головний екран користувач має можливість оглядати основні функції та взаємодіяти з навігацією додатку. Він може вибрати потрібну йому вкладку з меню навігації. Це групування полегшує роботу з додатком, спрощуючи логіку взаємодії з даними та забезпечуючи зручний інтерфейс для користувача.

На рисунку 3.6 зображено реалізацію навігації по меню.

```

<div class="primary-menu-container buddyx-desktop-menu">
  <ul id="primary-menu" class="menu">
    <li
      class="menu-item menu-item-type-custom menu-item-object-custom current-menu-item current_
      <a href="https://andrewcool.pp.ua/" aria-current="page">Home</a></li>
    <li class="menu-item menu-item-type-post_type menu-item-object-page menu-item-40"><a
      href="https://andrewcool.pp.ua/groups/">Groups</a></li>
    <li class="menu-item menu-item-type-post_type menu-item-object-page menu-item-41"><a
      href="https://andrewcool.pp.ua/members/">Members</a></li>
    <li
      class="menu-item menu-item-type-post_type menu-item-object-page menu-item-home menu-item-
      <a href="https://andrewcool.pp.ua/">Activity</a></li>
    <li class="menu-item menu-item-type-post_type menu-item-object-page menu-item-56"><a
      href="https://andrewcool.pp.ua/news/">News</a></li>
  </ul>
</div>

```

Рисунок 3.6 – Реалізація навігації

На головній сторінці однією з ключових функцій є перегляд стрічки активностей спільноти, яка пропонується в додатку. Ця стрічка активностей відображає найсвіжіші події та дії учасників спільноти, що дозволяє користувачам бути в курсі всього, що відбувається в їхньому середовищі.

Перегляд стрічки активностей на головній сторінці додатку є важливою функцією, яка допомагає користувачам залишатися на зв'язку та відчувати себе частиною спільноти. Ця стрічка відображає різноманітні події та дії, такі як нові повідомлення, коментарі, лайки, а також сповіщення про оновлення профілів та інші активності.

На рисунку 3.7 та 3.8 зображено реалізацію стрічки активностей.

```

<div id="activity-stream" class="activity" data-bp-list="activity">
  <div id="bp-ajax-loader">
    <aside class="bp-feedback bp-messages loading">
      <span class="bp-icon" aria-hidden="true"></span>
      <p>Loading the community updates. Please wait.</p>
    </aside>
  </div>
</div><!-- .activity -->

```

Рисунок 3.7 – Реалізація стрічки активностей

```
function bp_activity_render_latest_activities_block( $attributes = array() ) {
    $block_args = bp_parse_args(
        $attributes,
        array(
            'title'          => '',
            'maxActivities' => 5,
            'type'           => array( 'activity_update' ),
            'postId'         => 0,
        )
    );

    $max_activities = (int) $block_args['maxActivities'];

    if ( ! $block_args['postId'] ) {
        $block_args['postId'] = get_the_ID();
    }

    if ( ! $block_args['title'] ) {
        $block_args['title'] = __( 'Latest updates', 'buddypress' );
    }

    $member_id = 0;
    if ( $block_args['postId'] ) {
        $member_id = (int) get_post_field( 'post_author', $block_args['postId'] );
    } else {
        $member_id = bp_displayed_user_id();
    }
}
```

Рисунок 3.8 – Реалізація стрічки активностей

Під блоком активностей користувач має змогу залишити коментар просто натиснувши на кнопку "Додати коментар". Це дозволяє користувачам висловлювати свої думки, враження та реакції на події, які вони бачать у стрічці активностей.

Коментарі можуть містити текстові повідомлення, емодзі, посилання або навіть мультимедійні вкладення, що дозволяє користувачам більш ефективно виражати свої думки та реакції (див. рисунок 3.9).

```

case 'comment-reply' :
    if ( empty( $bp->activity->track ) ) {
        $bp->activity->track = bp_activity_get_post_types_tracking_args();
    }

    if ( ! empty( $bp->activity->track[ $activity_type ] ) ) {
        if ( isset( $bp->activity->track[ $activity_type ]->activity_comment ) ) {
            $retval = $bp->activity->track[ $activity_type ]->activity_comment;
        }

        if ( isset( $bp->activity->track[ $activity_type ]->comments_tracking ) ) {
            $retval = $bp->activity->track[ $activity_type ]->comments_tracking && ! bp_disable_blogforum_comments();
        }
    }
} elseif ( 'new_forum_topic' === $activity_type || 'new_forum_post' === $activity_type ) {
    $retval = ! bp_disable_blogforum_comments();
} else {
    $retval = true;
}
break;

```

Рисунок 3.9 – Реалізація коментарів

Коли користувач переходить до свого профілю в додатку, перед ним відкривається повний спектр функціоналу, який надається веб додатком. Цей функціонал включає в себе різноманітні можливості, такі як "Налаштування профілю", "Сповідання", "Приватні повідомлення", "Друзі", "Групи", "Медіа", "Розширені налаштування". На рисунку 3.10 зображено реалізацію переходу до цих функцій після переходу з головного меню.

```

<div class="site-wrapper member-home">
    <div class="bp-wrap">

        <nav class="main-navs no-ajax bp-navs single-screen-navs vertical users-nav"
            id="object-nav" role="navigation" aria-label="Member menu">
            <ul id="member-primary-nav" class="bp-priority-object-nav-nav-items">
                <li id="activity-personal-li" class="bp-personal-tab current selected">
                    <a href="https://andrewcool.pp.ua/members/woelian/activity/"
                        id="user-activity">
                        Activity
                    </a>
                </li>
                <li id="xprofile-personal-li" class="bp-personal-tab">
                    <a href="https://andrewcool.pp.ua/members/woelian/profile/"
                        id="user-xprofile">
                        Profile
                    </a>
                </li>
                <li id="notifications-personal-li" class="bp-personal-tab">
                    <a href="https://andrewcool.pp.ua/members/woelian/notifications/"
                        id="user-notifications">
                        Notifications
                        <span class="count">1</span>
                    </a>
                </li>
            </ul>
        </nav>
    </div>

```

Рисунок 3.10 – Реалізація функцій в профілі

Отже, було створено головний екран для веб додатку, а також навігаційне меню, яке дозволяє користувачам легко переходити до різних функцій програми. Цей підхід забезпечує зручний та ефективний процес ознайомлення з веб додатком та його використання. Додатково, наявність такого навігаційного меню робить навігацію більш інтуїтивно зрозумілою для користувачів, що сприяє покращенню їхнього досвіду використання додатку.

3.3 Розробка модуля передачі повідомлень

Модуль передачі повідомлень, є головним модулем системи, який слугує її ядром, забезпечуючи основну функціональність та інтеграцію інших модулів. Основні функції модуля передачі повідомлень включають передачу повідомлень між користувачами, забезпечення надійності доставки повідомлень, підтримку різних форматів даних та масштабованість.

Передача повідомлень між користувачами передбачає можливість відправки та отримання текстових, графічних і мультимедійних повідомлень. Це дозволяє користувачам ефективно обмінюватися різноманітною інформацією в межах системи. Надійність передачі повідомлень досягається завдяки використанню механізмів підтвердження отримання та повторної передачі у разі помилок, що забезпечує доставку повідомлень навіть у випадку тимчасових проблем з мережею або сервером. Підтримка різних форматів даних включає текстові, аудіо, відео та інші формати, що робить модуль універсальним та здатним задовольнити різноманітні потреби користувачів.

Масштабованість модуля передбачає його здатність підтримувати велику кількість одночасних користувачів та ефективно функціонувати при зростанні навантаження. Це досягається за рахунок оптимізації алгоритмів обробки повідомлень та використання розподілених обчислювальних ресурсів. Важливою складовою розробки модуля передачі повідомлень є інтеграція з іншими компонентами системи.

На рисунку 3.11 зображено блок схему роботи модуля передачі повідомлень.

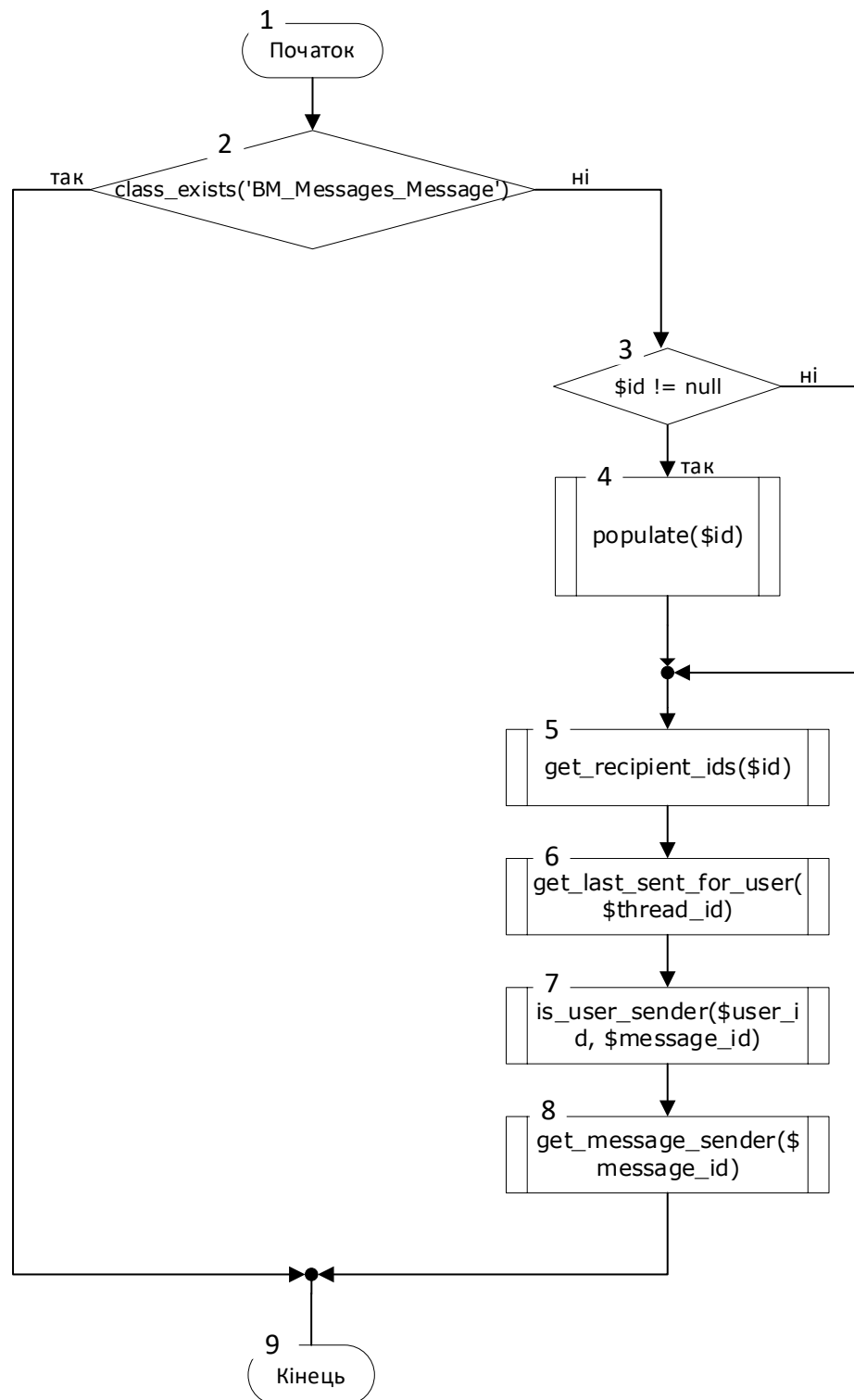


Рисунок 3.11 – Блок схема алгоритму роботи модуля передачі повідомлень

Крок 1. Початок роботи модуля.

Крок 2. Запуск циклу який перевіряє існування класу.

Крок 3-4. Запуск циклу який отримує всі id користувачів.

Крок 5. Отримання id отримувача.

Крок 6. Отримання останнього повідомлення від користувача до отримувача.

Крок 7. Перевірка чи є користувач отримувачем, якщо так то функція видасть помилку.

Крок 8. Відправлення повідомлення.

Крок 9. Завершення роботи модуля.

3.4 Розробка модуля інтегрування штучного інтелекту

Розробка модуля використання штучного інтелекту для аналізу емоційного стану користувача є одним із ключових завдань при створенні сучасних інтерактивних систем, оскільки він забезпечує глибше розуміння емоційних реакцій користувачів і підвищує якість їх взаємодії з системою. Модуль аналізу емоційного стану, що розглядається у цій роботі, є одним із головних модулів системи, який слугує її ядром, забезпечуючи основну функціональність та інтеграцію інших модулів. Для реалізації цього модуля використовується API OpenAI, що надає потужні інструменти для аналізу тексту та виявлення емоцій.

Основні функції модуля аналізу емоційного стану включають розпізнавання та класифікацію емоцій користувача на основі аналізу текстових повідомлень. Це дозволяє системі адаптувати свій інтерфейс і відповіді відповідно до емоційного стану користувача, підвищуючи рівень персоналізації та покращуючи користувацький досвід. Використання API OpenAI дозволяє інтегрувати алгоритми обробки природної мови, що забезпечують високу точність і ефективність аналізу.

Розпізнавання емоцій досягається шляхом надсилання текстових даних користувачів до API OpenAI, яке здійснює обробку та повертає емоційні

характеристики повідомлення. Це включає виділення основних емоційних категорій, таких як радість, сум, гнів, страх та нейтральний стан. OpenAI API використовує глибокі нейронні мережі та інші сучасні методи машинного навчання, що дозволяють точно ідентифікувати емоційний стан на основі контексту та лексичних характеристик тексту.

Інтеграція модуля аналізу емоційного стану з іншими компонентами системи забезпечує взаємодію різних модулів та підсистем, що підвищує загальну ефективність і функціональність системи. Використання API OpenAI дозволяє легко інтегрувати модуль у вже існуючу інфраструктуру системи, забезпечуючи гнучкість та можливість масштабування.

Масштабованість модуля передбачає його здатність підтримувати велику кількість одночасних користувачів та ефективно функціонувати при зростанні навантаження. Це досягається за рахунок оптимізації алгоритмів аналізу та використання розподілених обчислювальних ресурсів, що надаються OpenAI. API OpenAI також забезпечує стабільність та швидкодію сервісу, що є важливим для роботи в реальному часі.

Додатково до основних функцій модуль використання штучного інтелекту для аналізу емоційного стану користувача дозволяє здійснювати аналіз трендів у настроях користувачів з плином часу, що відкриває нові можливості для покращення сервісу. Наприклад, аналізуючи зміни в емоційному стані користувачів, система може визначати періоди підвищеного стресу або незадоволення і вчасно реагувати на ці зміни, пропонуючи відповідну підтримку чи зміни в інтерфейсі. Крім того, інтеграція з API OpenAI надає можливість для багатомовного аналізу, що дозволяє системі обробляти текстові дані користувачів, які спілкуються різними мовами. Це значно розширює аудиторію користувачів та підвищує глобальну конкурентоспроможність системи. Виявлення емоційних станів незалежно від мови користувача сприяє створенню більш інклюзивного та доступного середовища.

На рисунку 3.12 зображено блок схему роботи модуля передачі повідомлень.

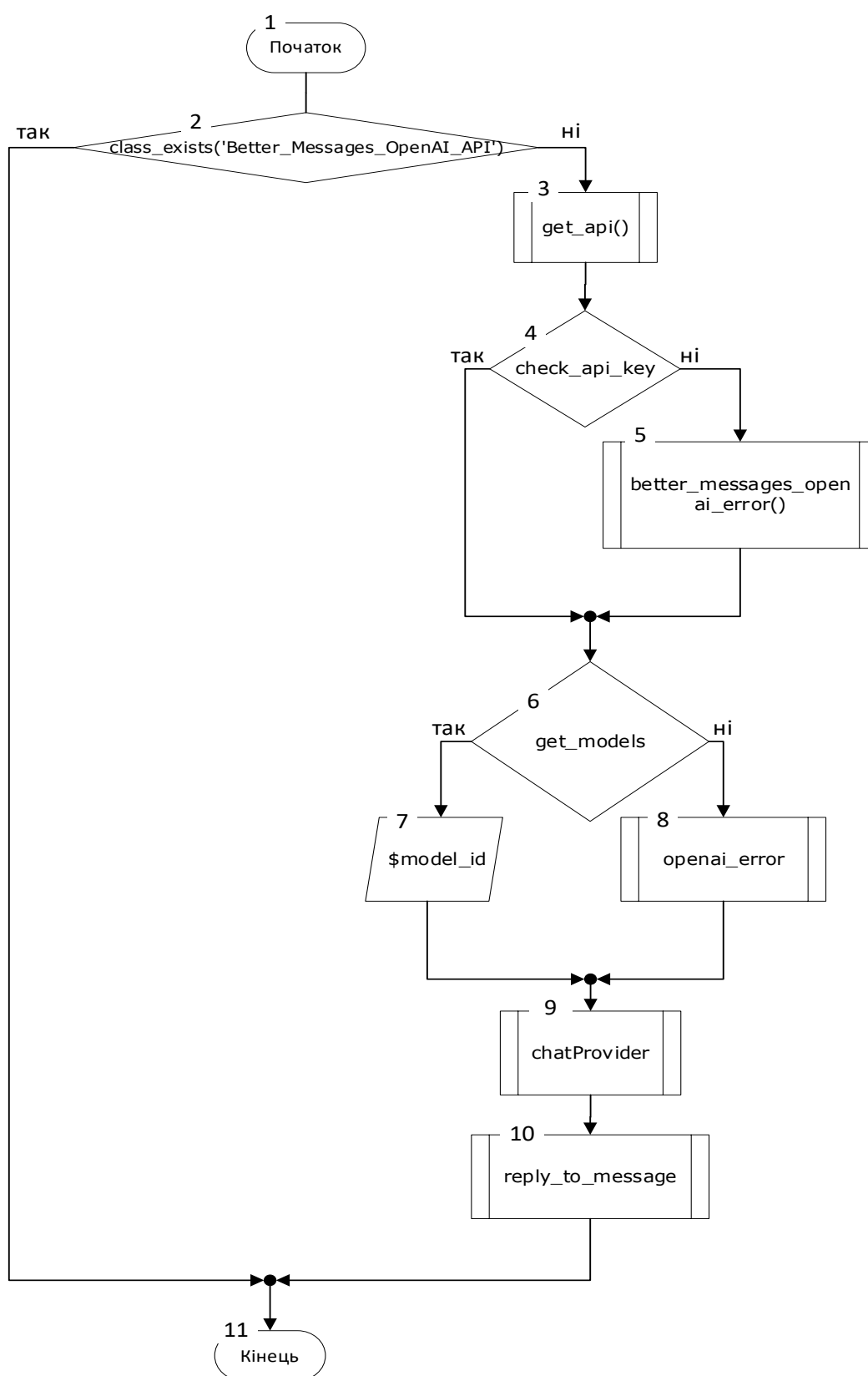


Рисунок 3.12 – Блок схема алгоритму роботи модуля інтеграції штучного інтелекту

Крок 1. Початок роботи модуля.

Крок 2. Запуск циклу який перевіряє існування класу.

Крок 3. Ввод ключа API.

Крок 4. Запуск перевірки ключа API.

Крок 5. Перевірка провалилася і йде вивод помилки.

Крок 6. Запуск алгоритму який отримує модель штучного інтелекту.

Крок 7. Ввод моделі штучного інтелекту.

Крок 8. Помилка в ввіді моделі штучного інтелекту.

Крок 9. Отримання повідомлення від користувача.

Крок 10. Відповідь штучного інтелекту на повідомлення

Крок 11. Завершення роботи модуля

3.5 Висновки

У третьому розділі розглянуто та обґрунтована вибірка інструментів для реалізації програмного продукту, технологій розробки та модулів. Під час аргументації вибору інструментів для створення програмного продукту було встановлено, що оптимальним рішенням для розробки додатку є використання мови PHP та системою керування вмістом WordPress.

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Тестування системи

У розробці програмного застосунку значну вагу приділяють тестуванню, що є важливим етапом [20]. Головна мета тестування полягає в виявленні можливих дефектів або помилок у програмному застосунку перед його випуском для користувачів, забезпечуючи таким чином його коректну роботу.

У тестуванні WordPress використовуються різноманітні методи для перевірки правильності та ефективності програмних продуктів. Серед них можна виділити наступні:

1. Тестування швидкості завантаження сторінки: Швидкість завантаження сторінки має велике значення для користувачів, оскільки вона впливає на їхнє враження від веб-сайту та користувацький досвід. Результати тестування дають розробникам уявлення про те, як швидко сторінка завантажується для реальних користувачів і які частини коду або ресурсів можуть бути оптимізовані для поліпшення швидкості завантаження (див. рисунок 4.1).

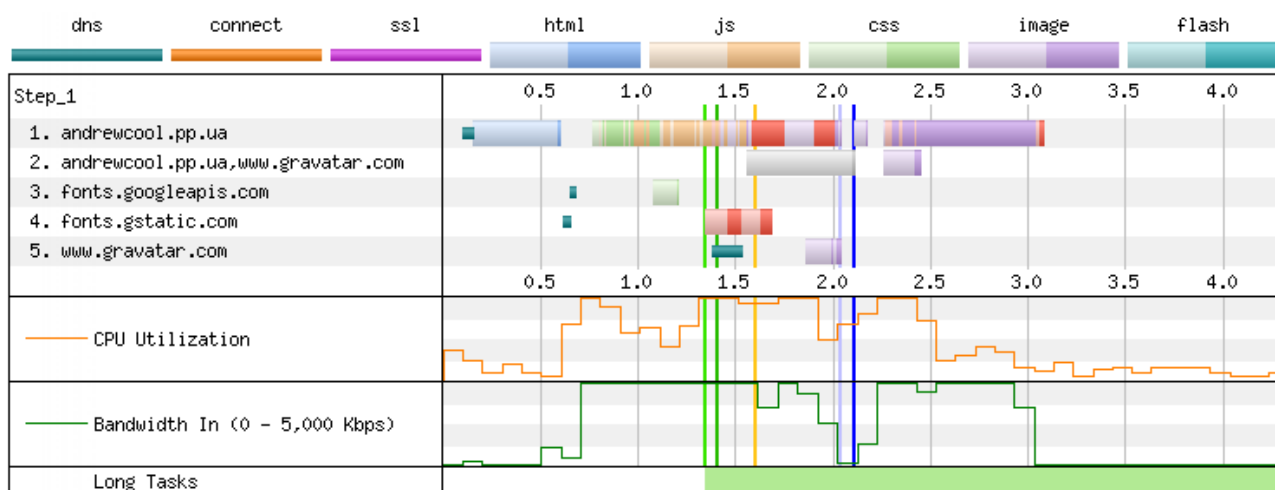


Рисунок 4.1 – Тестування продуктивності

2. Розбивка доменів: основна мета тесту на розбивку доменів полягає в тому, щоб збільшити паралелізм в процесі завантаження сторінок, оскільки

браузери обмежують кількість одночасних запитів до одного домену. Розподіл ресурсів між декількома піддоменами дозволяє більш ефективно використовувати можливості браузера і прискорює завантаження сторінок (див. рисунок 4.2).

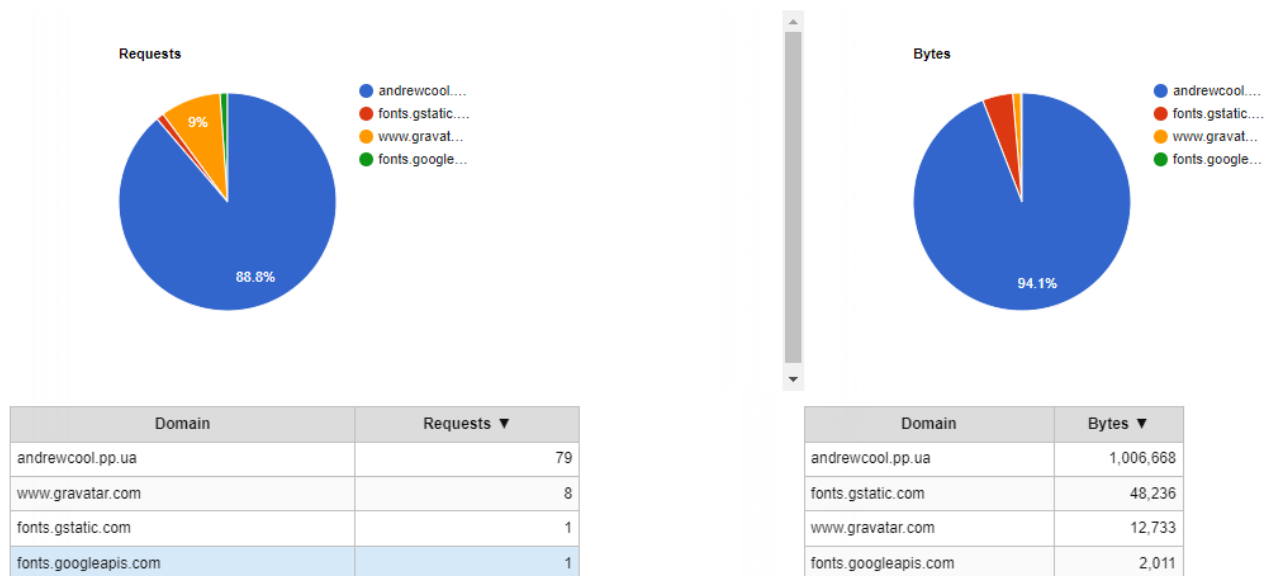


Рисунок 4.2 – Тестування розбивки доменів

Це дозволило переконатися, що програма працює ефективно та безперебійно навіть при великому навантаженні.

Після цього ми перевіряємо розділ "Забули пароль", вводячи порожні дані. Повідомлення про помилку важливо для уникнення непорозумінь та сприяє коректному використанню функціоналу "Забули пароль". Якщо користувач вводить невірну інформацію при відновленні пароля, повідомлення про помилку надає зрозумілу інструкцію про те, що необхідно виправити або додаткові кроки для відновлення доступу.

Цей механізм дозволяє покращити безпеку та забезпечити точність введених даних. Повідомлення про помилку допомагає уникнути непорозумінь та сприяє коректному використанню функціоналу "Забули пароль", забезпечуючи швидке та ефективне відновлення доступу до облікового запису для користувачів.

Результат тестування зображено на рисунку 4.3.

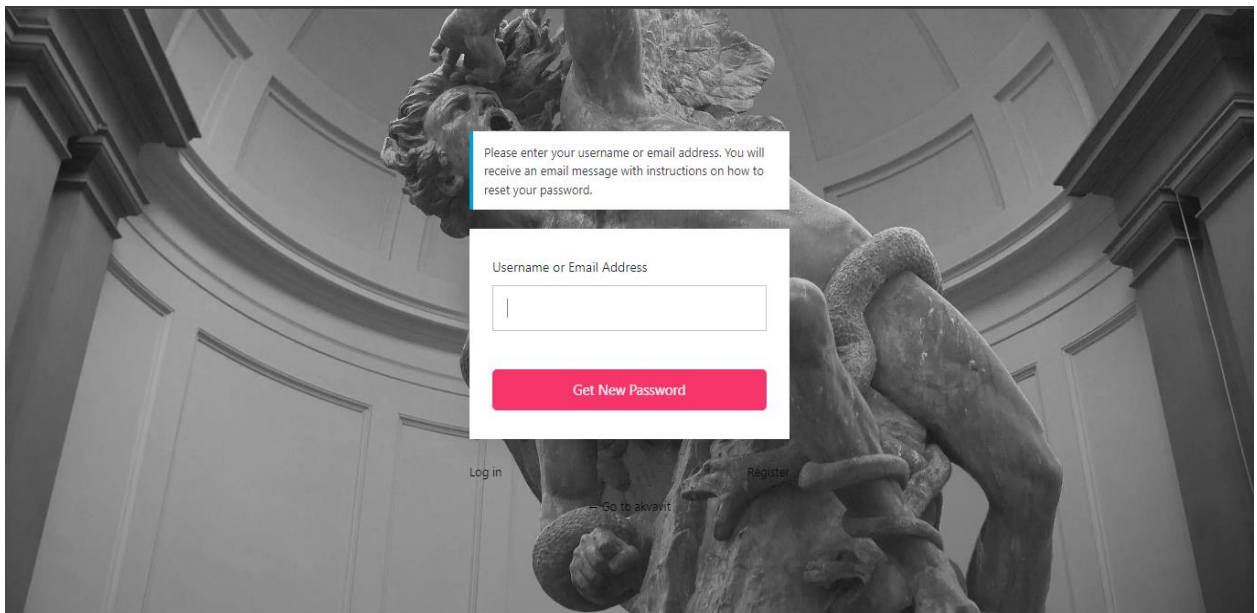


Рисунок 4.3 – Тестування спроби відновити пароль з пустими даними

Це підтверджує правильність реалізації даної функціональності та її здатність до надійного та безперебійного використання користувачами.

Тестування веб-сайту на різних браузерех є важливою частиною розробки, оскільки різні браузери можуть різнитися у підтримці стандартів веб-технологій та відображенні вмісту. Під час такого тестування розробники перевіряють, як сторінки відображаються та працюють в різних браузерех, включаючи Google Chrome, Mozilla Firefox, Microsoft Edge та інші. Основні аспекти тестування включають перевірку:

1. Відмінностей у вигляді: Перевірка, чи виглядає сайт однаково на різних браузерех, включаючи шрифти, кольори, розміри та розташування елементів.
2. Робота PHP: Впевненість, що функції працюють на всіх браузерех та немає помилок.
3. Підтримки HTML5 та CSS3: Впевненість, що нові функції HTML5 та CSS3 правильно відображаються та працюють на різних браузерех.

На рисунках 4.4, 4.5 та 4.6 показано результат тестування на коректне відображення та на виконання функціоналу

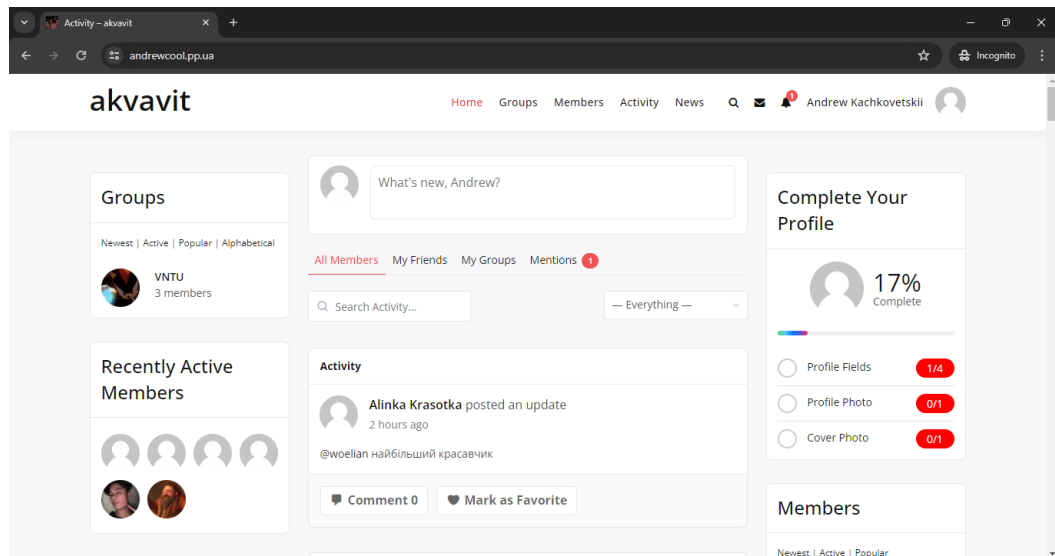


Рисунок 4.3 – Тестування в браузері ‘Google Chrome’

Тестування веб-додатку в Google Chrome було пройдено успішно, і були виявлені позитивні результати. Сайт відображається коректно, всі елементи розташовані на своїх місцях, а відмінності у вигляді між різними розділами сторінки мінімальні.

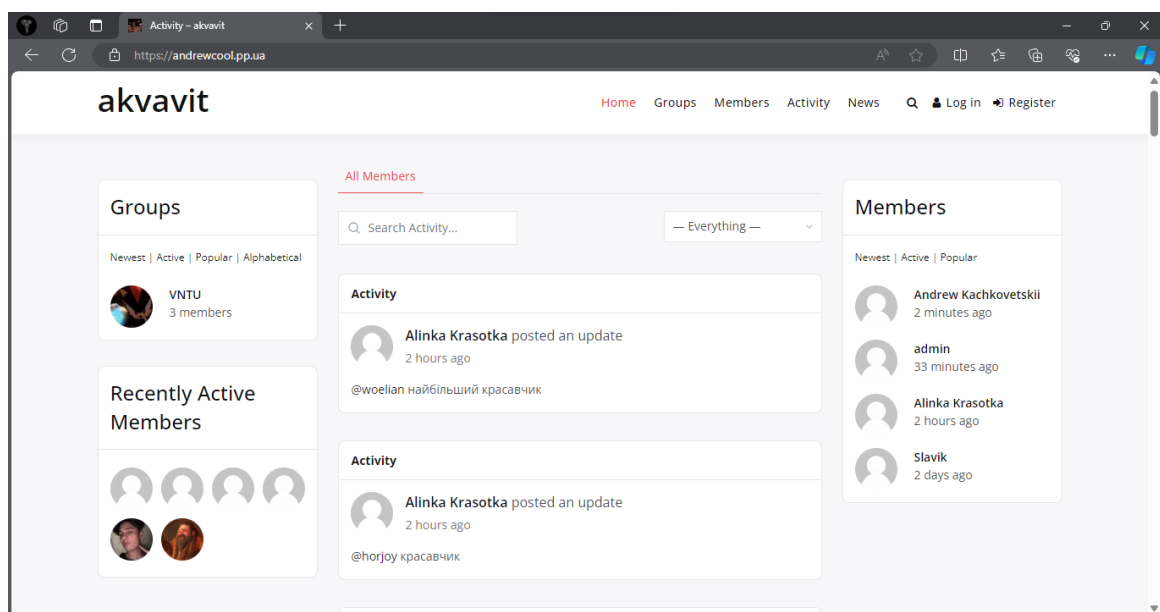


Рисунок 4.4 – Тестування в браузері ‘Microsoft Edge’

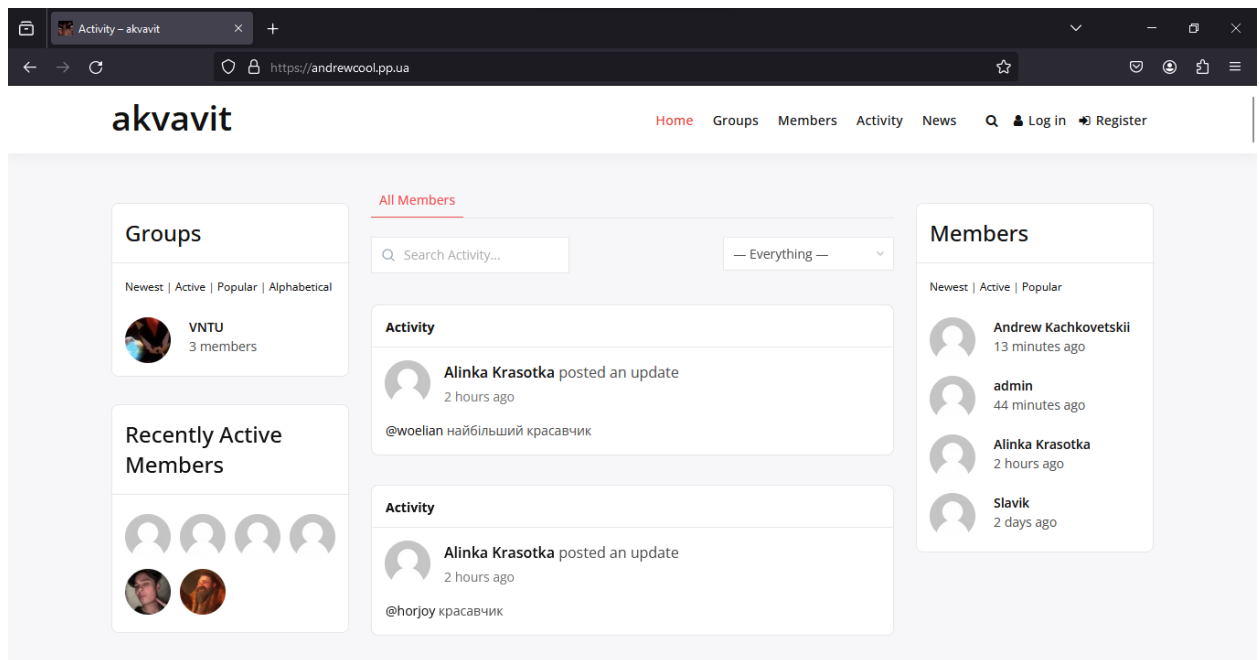


Рисунок 4.5 – Тестування в браузері 'Mozilla Firefox'

Тестування веб-сторінки у 3 найпопулярніших браузерах (Google Chrome, Mozilla Firefox, Microsoft Edge) пройшло успішно. Сайт відображається коректно і функціонує без помилок у всіх трьох браузерах, що підтверджує його сумісність з широким спектром веб-переглядачів. Усі елементи сторінки розташовані на своїх місцях, стилізація і кольори відповідають дизайну, а взаємодія з користувачем працює плавно та без затримок. JavaScript функції працюють правильно у всіх тестових браузерах, а відгуки на дотик реагують адаптивно і швидко.

Тестування веб-сторінки на відображення на мобільних пристроях включає перевірку її адаптивності, коректності відображення та функціональності на різних типах мобільних пристроїв. Під час тестування розробники перевіряють, як веб-сторінка виглядає на різних розмірах екрану, включаючи смартфони, планшети та інші мобільні пристрої. Основні аспекти тестування включають перевірку, чи адекватно відображається контент на малих екранах, чи коректно працюють елементи взаємодії, такі як кнопки та посилання, чи швидко та ефективно завантажується сторінка на мобільних мережах, та чи

виникають помилки або проблеми при використанні сторінки на мобільних пристроях.

Результати тестування відображення веб додатку на мобільних пристроях відображено на рисунку 4.6.

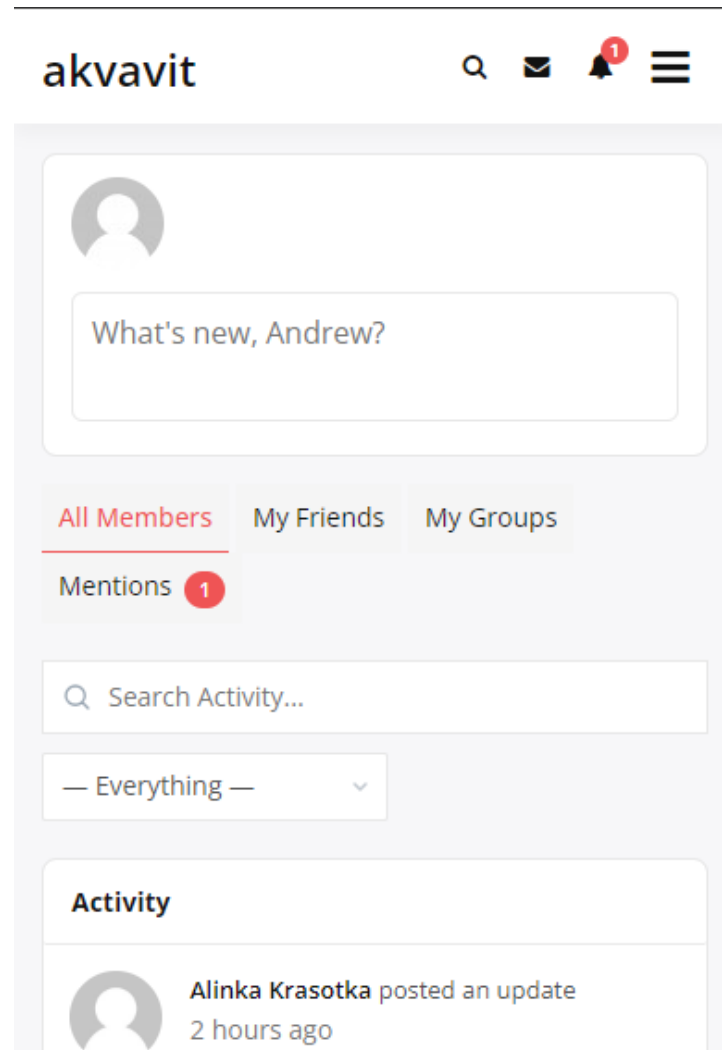


Рисунок 4.6 – Тестування відображення веб додатку на мобільних пристроях

Завершальним етапом тестування стало комплексне тестування, під час якого вже тестувалася вся система в цілому. Здійснювалися різні сценарії використання, що моделювали реальні умови користування програмою. Також важливим аспектом було тестування системи на різних пристроях та під різними операційними системами з метою виявлення можливих проблем з сумісністю.

Результати тестування наведено в таблиці 4.1.

Таблиця 4.1– Результати тестування

№	Опис тесту	Кроки виконання	Очікуваний результат	Фактичний результат	Оцінка
1	2	3	4	5	6
1	Створення нового посту	1. Натиснути кнопку "Створити пост"	Відображення форми створення посту	Відображення створеного посту	Успішно
		2. Ввести текст посту	Правильне введення тексту	Правильне введення тексту	
		3. Натиснути кнопку "Опублікувати"	Пост з'являється у стрічці користувача	Пост з'являється у стрічці користувача	
2	Редагування профілю користувача	1. Зайти в налаштування профілю	Відображення сторінки налаштувань профілю	Відображення сторінки налаштувань профілю	Успішно
		2. Внести зміни в інформацію профілю	Успішне внесення змін	Успішне внесення змін	
		3. Натиснути кнопку "Зберегти зміни"	Збереження внесених змін	Збереження внесених змін	

Продовження таблиці 4.1

1	2	3	4	5	6
3	Пошук користувача	1. Ввести ім'я користувача у поле пошуку	Відображення списку користувачів, які відповідають запиту	Відображення списку користувачів, які відповідають запиту	Успішно
		2. Вибрати потрібного користувача	Перехід на профіль обраного користувача	Перехід на профіль обраного користувача	
4	Перевірка штучного інтелекту на визначення емоційного стану	1. Ввести текст з різними емоційними відтінками	Аналіз тексту та визначення емоційного стану	Аналіз тексту та визначення емоційного стану	Успішно
		2. Отримати відповідь від штучного інтелекту	Персоналізована відповідь від ШІ	Персоналізована відповідь від ШІ	

Отже, тестування системи включає оцінку функціональності, надійності та продуктивності програмного забезпечення. Виконані тести охоплювали тестування додатку на швидкість завантаження, розбивки доменів, відновлення паролю, та тестування на коректну роботу додатку на різних браузерях та мобільних пристроях

4.2 Розробка інструкції користувача

Інструкція користувача пропонує послідовні кроки, необхідні для досягнення певних цілей або виконання операцій. Основна мета цієї інструкції - забезпечити зручність та ефективність у використанні продукту. Деталі щодо мінімальної конфігурації браузеру користувача наведено в таблиці 4.1.

Таблиця 4.2 – Мінімальна вимоги:

Тип браузера	Google Chrome
Тип процесора	2-ядерний з тактовою частотою не менше 1.5 ГГц
Об'єм оперативної пам'яті	2 ГБ
Місце на жорсткому диску	100 Мб
Версія Google Chrome	7.0.517.44
Пропускна здатність мережі	10 Mbps

Деталі щодо рекомендованих вимог до браузеру наведено в таблиці 4.3.

Таблиця 4.3 – Рекомендовані вимоги:

Тип браузера	Google Chrome
1	2
Тип процесора	4-ядерний з тактовою частотою 2.0 ГГц або вище
Об'єм оперативної пам'яті	6 ГБ

Продовження таблиці 4.3

1	2
Місце на жорсткому диску	200 Mб
Версія Google Chrome	124.0.6367.92
Пропускна здатність мережі	50 Mbps

Під час процедури авторизації в системі будь-які помилки негайно повідомляються користувачу, щоб він міг виправити їх та продовжити процес. Це забезпечує зручність та швидкість взаємодії з системою.

Після успішної авторизації система перевіряє наявність облікового запису. У випадку, якщо користувач забув свій пароль та намагається скинути його, не вказавши електронної пошти, або якщо такого облікового запису не існує в системі, він отримає повідомлення про помилку та система запросить правильні дані від користувача. Ця функціональність сприяє коректному використанню функціоналу "Забули пароль" і забезпечує безпеку облікових записів користувачів.

Крім того, негайне повідомлення про помилку дозволяє уникнути непорозумінь та швидко вирішувати проблеми, що можуть виникнути в процесі авторизації та відновлення доступу.

Отже, інструкція користувача виступає ключовим документом, який сприяє успішному використанню продукту чи послуги, надаючи користувачам необхідну інформацію та вказівки для ефективної взаємодії з ними. Вона допомагає уникнути непорозумінь і помилок, що дозволяє заощадити час користувача, оскільки містить чітку та структуровану інформацію.

4.3 Висновки

У четвертому розділі проведено тестування функціональних та графічних елементів застосунку для перевірки їх працездатності. На основі результатів тестування було складено керівництво користувача для системи, яке містить вичерпну інформацію про конфігурацію пристрою користувача.

ВИСНОВКИ

У ході бакалаврської дипломної роботи розроблено інтерактивну комунікаційну систему передачі повідомлень. В процесі виконання роботи розв'язані такі задачі проведено аналіз існуючих методів і засобів систем передачі повідомлень для визначення напрямків підвищення ефективності та зручності обміну інформацією між користувачами, запропоновано метод аналізу настроїв та емоційного стану користувачів, розроблено програмні компоненти та інтерактивну комунікаційну систему передачі повідомлень на основі запропонованих методів, проведено експериментальні дослідження розроблених засобів..

У процесі досліджень використано такі теорії та методи: методи лінгвістичного аналізу, методи синтаксичного та семантичного аналізу, методи машинного навчання для класифікації та кластеризації даних, методи візуалізації для аналізу взаємодій, методи моделювання даних, комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

Подальшого розвитку отримав метод аналізу настроїв та емоційного стану користувачів, у якому, на відміну від існуючих, використано штучний інтелект для аналізу емоційного тону повідомлень, що дозволило виявляти настрої користувачів, відповідним чином реагувати, і, як наслідок, зменшити емоційне навантаження при спілкуванні.

У першому розділі детально розглянуто різні аспекти розробки веб застосунку інтерактивної системи передачі повідомлень. Почавши з аналізу потреб та очікувань користувачів, виявились основні функціональні потреби, які повинен вирішувати додаток. Дослідження ринку дозволило виявити наявні системи та їх переваги та недоліки, що видалося надзвичайно корисним у формулюванні власних стратегій.

У другому розділі проведено аналіз вхідних та вихідних даних програмного застосунку для визначення необхідного функціоналу та потреб користувачів. Алгоритм роботи вебзастосунку розроблено з урахуванням потреб

користувачів та особливостей функціоналу. Розроблено модель роботи програмного продукту за допомогою UML діаграми. Модель надає чітке уявлення про функціонування застосунку у реальному середовищі.

У третьому розділі розглянуто та обґрунтована вибірка інструментів для реалізації програмного продукту, технологій розробки та модулів. Під час аргументації вибору інструментів для створення програмного продукту було встановлено, що оптимальним рішенням для розробки додатку є використання мов HTML, CSS, PHP та системою керування вмістом WordPress.

У четвертому розділі проведено тестування функціональних та графічних елементів застосунку для перевірки їх працездатності. На основі результатів тестування було складено керівництво користувача для системи, яке містить вичерпну інформацію про конфігурацію пристрою користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Качковецький А. І. Перспективи Інтерактивних Систем Передачі Повідомлень // Матеріали Науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії, Вінниця, 2024. [Електронний ресурс] Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/view/21562>
2. Social Network Sites and Its Popularity [Електронний ресурс] – Режим доступу до ресурсу: <https://www.proquest.com/openview/> (дата звернення 30.02.2024).
3. Безпечне кодування - Виклики інформаційної безпеки [Електронний ресурс] – Режим доступу до ресурсу: <https://www.dqsglobal.com/uk-ua/navchajtesya/blog/bezpechne-koduvannya-vikliki-informacijnoyi-bezpeki> (дата звернення ..).
4. We are social 2023: соціальні мережі, інтернет та тенденції електронної комерції [Електронний ресурс] – Режим доступу до ресурсу: <https://elit-web.ua/ua/blog/we-are-social-2023> (дата звернення ..).
5. Поради та Практичні Рекомендації [Електронний ресурс] – Режим доступу до ресурсу: <https://bdut.co.ua/pro-nas/rozrobka-mobilnyh-dodatkov/> (дата звернення: 08.03.2024)
6. Аналіз конкурентів вашого додатку [Електронний ресурс] – Режим доступу до ресурсу: <https://asomobile.net/uk/blog/analiz-konkurentiv-vashogo-dodatku-novi-instrumenti-vid-asomobile/> (дата звернення: 08.03.2024)
7. China Tech 101: QQ [Електронний ресурс] – Режим доступу до ресурсу: <https://www.chinafy.com/china-tech/what-is-qq-and-how-to-get-started> (дата звернення ..).
8. Життя без реклами: Історія створення та розвитку WhatsApp [Електронний ресурс] – Режим доступу до ресурсу: <https://investory.news/zhittya-bez-reklami-istoriya-stvorenniya-ta-rozvitku-whatsapp/> (дата звернення ..).

9. Facebook social network [Електронний ресурс] – Режим доступу до ресурсу: <https://www.britannica.com/money/Facebook> (дата звернення ..).
10. What is Myspace? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/what-is-myspace> (дата звернення ..).
11. 533 million Facebook users' phone numbers and personal data have been leaked online [Електронний ресурс] – Режим доступу до ресурсу: <https://www.businessinsider.com/stolen-data-of-533-million-facebook-users-leaked-online-2021-4> (дата звернення ..).
12. Специфіка розробки UX/UI веб-дизайну [Електронний ресурс] – Режим доступу до ресурсу: <https://rubarbs.com/ua/article/the-specifics-of-the-development-of-ux-ui-web-design> (дата звернення ..).
13. The Importance of Your Site's Homepage [Електронний ресурс] – Режим доступу до ресурсу: <https://www.keywebconcepts.com/the-importance-of-your-sites-homepage/> (дата звернення ..).
14. Авторизація та особистий кабінет користувача [Електронний ресурс] – Режим доступу до ресурсу: <https://design.gov.ua/ua/proektirovaie-i-prototipirovanie/oblikovij-zapis-koristuvacha> (дата звернення ..).
15. James Kalbach. (2007). Designing Web Navigation: Optimizing the User Experience 1st Edition. – 34 ст.
16. Jon Duckett. (2011). HTML and CSS: Design and Build Websites First Edition. – 59 ст.
17. Kevin Tatroe. (2020). Programming PHP: Creating Dynamic Web Pages 4th Edition. – 106 ст.
18. What Is WordPress? Explained for Beginners [Електронний ресурс] – Режим доступу до ресурсу: <https://kinsta.com/knowledgebase/what-is-wordpress/> (дата звернення ..).
19. Переваги дизайну UI [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mindseo.com/resource/user-interface-design/> (дата звернення: 20.03.2024).

20. Методичні вказівки до виконання курсових проєктів з дисципліни «Проектний практикум» для студентів спеціальності 121 «Інженерія програмного забезпечення» / Уклад. В.В. Войтко, Г. О. Черноволик – Вінниця: ВНТУ, 2022. – 44 с.

ДОДАТКИ

Додаток А – Технічне завдання

63

Додаток А – Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., професор
О. Н. Романюк
«12» березня 2024 р.

Технічне завдання
на бакалаврську дипломну роботу «Розробка інтерактивної
комунікаційної системи передачі повідомлень» за спеціальністю 121 –
Інженерія програмного забезпечення

Керівник бакалаврської дипломної роботи:
 к.т.н., доцент О. М. Рейда
«12» березня 2024 р.

Виконав:
 студент гр. ЗПП-206 А. І. Качковецький
«12» березня 2024 р.

Вінниця – 2024 року

1. Найменування та галузь застосування

Бакалаврська дипломна робота: «Розробка інтерактивної комунікаційної системи передачі повідомлень».

Галузь застосування – заклади освіти.

2. Підстава для розробки

Підставою для виконання бакалаврської дипломної роботи (БДР) є індивідуальне завдання на БДР та наказ №80 від «11» березня 2024 р. ректора ВНТУ про закріплення тем БДР.

3. Мета та призначення розробки

Метою роботи є підвищення ефективності та зручності обміну інформацією між користувачами, а також спрямованна на створення сприятливого середовища для взаємодії користувачів. Що сприятиме поліпшенню досвіду передачі інформації між користувачами.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись БДР.

1. А.М. Петух, О.В. Романюк, О.Н. Романюк. Бази даних. Мови запитів, управління транзакціями, розподілена обробка даних. Вінниця : ВНТУ, 2016. 97 с.
2. O'Regan Gerard. Concise Guide to Software Testing. Хам, Швейцарія : Springer International Publishing, 2019. 293 с

5. Технічні вимоги

Тип програми – комп'ютериний застосунок; модель розробки каскадна; система управління змістом – WordPress; вхідні дані: персональні дані входу користувачів; вихідні дані: відображення публічних та приватних повідомлень користувачів; засоби розробки – MS Visual Studio Code; мова програмування – PHP.

6. Конструктивні вимоги

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

1. Пояснювальна записка до БДР.
2. Технічне завдання.
3. Лістинги програми

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ

9. Стадії та етапи розробки:

№ п\п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи
1	Аналіз інтерактивних системи передачі повідомлень	13.03.24 – 22.03.24
2	Розробка інтерфейсу та моделей системи	13.03.24 – 22.03.24
3	Розробка програмних модулів реалізації вебдодатку	25.03.24 – 19.04.24
4	Тестування вебдодатку інтерактивної системи передачі повідомлень	22.04.24 – 17.05.24
5	Оформлення матеріалів до захисту БДР	20.05.24 – 03.06.24

10. Порядок контролю та прийняття

Виконання етапів бакалаврської дипломної роботи контролюється керівником згідно з графіком виконання роботи. Прийняття бакалаврської дипломної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Додаток Б – Протокол перевірки на плагіат

66

Додаток Б – Протокол перевірки на плагіат

ПРОТОКОЛ ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка інтерактивної комунікаційної системи передачі повідомлень.

Тип роботи: БДР

Підрозділ: кафедра програмного забезпечення, ФІТКІ

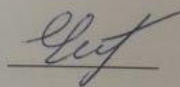
Науковий керівник: Рейда О. М.

Оригінальність	88,1%
Схожість	11,9%

Аналіз звіту подібності

- Запозичення, виявлені у роботі, оформленні коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цілісності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховання недобросовісних запозичень.

Особа, відповідальна за перевірку



Черноволик Г.О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichек

Автор роботи

Керівник роботи



Качковецький А. І.

Рейда О. М.

Додаток В – Лістинг програми

```

Openai.php

<?php
use BetterMessages\React\EventLoop\Loop;
use BetterMessages\React\Http\Browser;
use BetterMessages\React\Stream\ThroughStream;

if( ! defined( 'ABSPATH' ) ) exit;

if ( ! class_exists( 'Better_Messages_OpenAI_API' ) ) {
    class Better_Messages_OpenAI_API
    {
        public static function instance()
        {

            static $instance = null;

            if (null === $instance) {
                $instance = new Better_Messages_OpenAI_API();
            }

            return $instance;
        }

        public function get_api()
        {
            $api_key = Better_Messages()->settings['openAiApiKey'];
            return BetterMessages_OpenAI::client($api_key);
        }

        public function check_api_key()
        {
            $api = $this->get_api();

            try {
                $api->models()->list();
                delete_option('better_messages_openai_error');
                return true;
            } catch (Exception $e) {
                update_option( 'better_messages_openai_error', $e->getMessage(),
false );
                return $e->getMessage();
            }
        }
    }
}

```

```

    }
}

public function get_models()
{
    $api = $this->get_api();

    try{
        $response = $api->models()->list();

        $models = [];

        foreach ($response->data as $result) {
            $model_id = $result->id;
            if( str_starts_with( $model_id, 'gpt') && ! str_contains(
$model_id, 'vision' ) ){
                $models[] = $model_id;
            }
        }

        return $models;
    } catch (Exception $e) {
        return new WP_Error( 'openai_error', $e->getMessage() );
    }
}

function chatProvider( $bot_id, $bot_user, $message ) {
    global $wpdb;

    $client = $this->get_api();

    $bot_settings = Better_Messages()->ai->get_bot_settings( $bot_id );

    $bot_user_id = absint( $bot_user->id ) * -1;

    $messages = $wpdb->get_results( $wpdb->prepare(
"SELECT id, sender_id, message
FROM ``.bm_get_table('messages') . ``
WHERE thread_id = %d
AND created_at <= %d
ORDER BY `created_at` ASC ", $message->thread_id, $message-
>created_at ) );

```

```

$request_messages = [];

if( ! empty( $bot_settings['instruction'] ) ) {
    $request_messages[] = [
        'role' => 'system',
        'content' => apply_filters(
'better_messages_open_ai_bot_instruction', $bot_settings['instruction'], $bot_id,
$message->sender_id )
    ];
}

foreach ( $messages as $_message ){
    $is_error = Better_Messages()->functions->get_message_meta(
$_message->id, 'ai_response_error' );
    if( $is_error ) continue;

    $content = [];

    $content[] = [
        'type' => 'text',
        'text' => preg_replace('/<!--(.\s)*?-->/', "", $_message->message)
    ];

    if( $bot_settings['support_images'] ) {
        $attachments = Better_Messages()->functions-
>get_message_meta($_message->id, 'attachments', true);

        if (!empty($attachments)) {
            foreach ( $attachments as $attachment ) {
                $url = $attachment;

                if (defined('BM_DEV')) {
                    $url = 'https://www.wordplus.org/wp-
content/uploads/2023/01/preview.jpg';
                }

                $content[] = [
                    "type" => "image_url",
                    "image_url" => ["url" => $url]
                ];
            }
        }
    }
}

```

```

        $request_messages[] = [
            'role' => $message->sender_id === $bot_user_id ? 'assistant' :
'user',
            'content' => $content,
        ];
    }

    try{
        $stream = $client->chat()->createStreamed([
            'model' => $bot_settings['model'],
            'messages' => $request_messages,
            'user' => $message->sender_id
        ]);

        foreach($stream as $response){
            $text = $response->choices[0]->delta->content;
            yield $text;
        }

        yield ['meta', json_encode($stream->meta())];
    } catch (Exception $e) {
        yield ['error', $e->getMessage()];
    }
}

function process_reply( $bot_id, $message_id )
{

    if( wp_get_scheduled_event(
'better_messages_ai_bot_ensure_completion', [ $bot_id, $message_id ] ) ){
        wp_clear_scheduled_hook(
'better_messages_ai_bot_ensure_completion', [ $bot_id, $message_id ] );
    }

    $message = Better_Messages()->functions->get_message( $message_id
);

    if( ! $message ){
        return;
    }

    if( empty( Better_Messages()->functions->get_message_meta(
$message_id, 'ai_waiting_for_response' ) ) ){
        return;
    }

```

```

    }

    $recipient_user_id = $message->sender_id;

    $bot_user = Better_Messages()->ai->get_bot_user( $bot_id );

    if( ! $bot_user ){
        return;
    }

    $ai_user_id = absint( $bot_user->id ) * -1;
    $ai_thread_id = $message->thread_id;

    $ai_message_id = Better_Messages()->functions->get_message_meta(
$message_id, 'ai_response_id' );

    if( $ai_message_id ){
        $ai_message = Better_Messages()->functions->get_message(
$ai_message_id );
        if( ! $ai_message ){
            $ai_message_id = false;
        }
    }

    if( ! $ai_message_id ){
        $ai_message_id = Better_Messages()->functions->new_message([
            'sender_id' => $ai_user_id,
            'thread_id' => $ai_thread_id,
            'content'    => '...',
            'return'     => 'message_id',
            'error_type' => 'wp_error'
        ]);

        Better_Messages()->functions->add_message_meta(
$ai_message_id, 'ai_response_for', $message_id );

        if( ! is_wp_error( $ai_message_id ) ){
            Better_Messages()->functions->add_message_meta(
$ai_message_id, 'ai_response_start', time() );
            Better_Messages()->functions->add_message_meta( $message_id,
'ai_response_id', $ai_message_id );
        } else {
            return;
        }
    }

```

```

    }

    $loop = Loop::get();

    $browser = new Browser($loop);

    $dataProvider = $this->chatProvider( $bot_id, $bot_user, $message );

    $stream = new ThroughStream(function ($data) {
        return $data;
    });

    $parts = [];

    $loop->addPeriodicTimer(0.1, function ($timer) use ($loop, $stream,
    $dataProvider, $message_id, $ai_user_id, $ai_message_id, $ai_thread_id, &$parts,
    $recipient_user_id) {
        if ( $dataProvider->valid() ) {
            $part = $dataProvider->current();

            if( is_array($part) && $part[0] === 'error' ){
                $stream->write( $part[1] );
                $stream->end();
                $loop->cancelTimer($timer);
                $loop->stop();

                $args = [
                    'sender_id' => $ai_user_id,
                    'thread_id' => $ai_thread_id,
                    'message_id' => $ai_message_id,
                    'content' => $part[1]
                ];

                Better_Messages()->functions->update_message( $args );

                Better_Messages()->functions->delete_message_meta(
                $message_id, 'ai_waiting_for_response' );
                Better_Messages()->functions->delete_thread_meta(
                $ai_thread_id, 'ai_waiting_for_response' );
                do_action( 'better_messages_thread_self_update', $ai_thread_id,
                $recipient_user_id );
                do_action( 'better_messages_thread_updated', $ai_thread_id,
                $recipient_user_id );
            }
        }
    });

```

```

        Better_Messages()->functions->add_message_meta(
$ai_message_id, 'ai_response_error', $part[1] );
        Better_Messages()->functions->add_message_meta(
$message_id, 'ai_response_error', $part[1] );
        return;
    }

    if( is_array($part) && $part[0] === 'meta' ){
        $stream->end();
        $loop->cancelTimer($timer);
        $loop->stop();

        $args = [
            'sender_id' => $ai_user_id,
            'thread_id' => $ai_thread_id,
            'message_id' => $ai_message_id,
            'content' => htmlentities( implode(" ", $parts) )
        ];

        Better_Messages()->functions->update_message( $args );

        Better_Messages()->functions->update_message_meta(
$message_id, 'openai_meta', $part[1] );
        Better_Messages()->functions->update_message_meta(
$message_id, 'ai_response_finish', time() );
        Better_Messages()->functions->delete_message_meta(
$message_id, 'ai_waiting_for_response' );
        Better_Messages()->functions->delete_thread_meta(
$ai_thread_id, 'ai_waiting_for_response' );
        do_action( 'better_messages_thread_self_update', $ai_thread_id,
$recipient_user_id );
        do_action( 'better_messages_thread_updated', $ai_thread_id,
$recipient_user_id );
        return;
    }

    $parts[] = $part;
    $stream->write( $part );
    $dataProvider->next();
} else {
    $stream->end();
    $loop->cancelTimer($timer);
    $loop->stop();
}

```

```

    }
});

if( Better_Messages()->websocket ) {
    $socket_server = apply_filters('bp_better_messages_realtime_server',
'https://realtime-cloud.bpbettermessages.com/');
    $bm_endpoint = $socket_server . 'streamMessage';

    $browser->post($bm_endpoint, [
        'x-site-id' => Better_Messages()->websocket->site_id,
        'x-secret-key' => sha1(Better_Messages()->websocket->site_id .
Better_Messages()->websocket->secret_key),
        'x-message-id' => $ai_message_id,
        'x-thread-id' => $ai_thread_id,
        'x-recipient-user-id' => $recipient_user_id,
        'x-sender-user-id' => $ai_user_id,
    ], $stream);
}

$loop->run();
}
public function reply_to_message( WP_REST_Request $request )
{
    ignore_user_abort(true);
    set_time_limit(60);

    $bot_id    = (int) $request->get_param( 'bot_id' );
    $message_id = (int) $request->get_param( 'message_id' );

    $this->process_reply( $bot_id, $message_id );
}
}
}

```

Messege.php

```
<?php
if( ! class_exists( 'BM_Messages_Message' ) ):
    /**
     * Single message class.
     */
    class BM_Messages_Message {
        /**
         * ID of the message.
         *
         * @var int
         */
        public $id;

        /**
         * ID of the message thread.
         *
         * @var int
         */
        public $thread_id;

        /**
         * ID of the sender.
         *
         * @var int
         */
        public $sender_id;

        /**
         * Subject line of the message.
         *
         * @var string
         */
        public $subject;

        /**
         * Content of the message.
         *
         * @var string
         */
        public $message;

        /**
         * Date the message was sent.
```

```

*
* @var string
*/
public $date_sent;

public $created_at;

public $updated_at;

public $temp_id;

/**
 * Message recipients.
 *
 * @var bool|array
 */
public $recipients = false;

public $count_unread = true;

/**
 * Constructor.
 *
 * @param int|null $id Optional. ID of the message.
 */
public function __construct( $id = null ) {
    $this->date_sent = bp_core_current_time();
    $this->sender_id = Better_Messages()->functions-
>get_current_user_id();

    if ( ! empty( $id ) ) {
        $this->populate( $id );
    }
}

/**
 * Set up data related to a specific message object.
 *
 * @param int $id ID of the message.
 */
public function populate( $id ) {
    global $wpdb;

```

```

        if ( $message = $wpdb->get_row( $wpdb->prepare( "SELECT *
FROM " . $table . " WHERE id = %d", $id ) ) ) {
            $this->id      = (int) $message->id;
            $this->thread_id = (int) $message->thread_id;
            $this->sender_id = (int) $message->sender_id;
            $this->message   = $message->message;
            $this->date_sent = $message->date_sent;
            $this->created_at = $message->created_at;
            $this->updated_at = $message->updated_at;
        }
    }

    /**
     * Send a message.
     *
     * @return int|bool ID of the newly created message on success, false on
failure.
     */
    public function send() {
        global $wpdb;

        $this->sender_id = apply_filters(
'better_messages_message_sender_id_before_save', $this->sender_id, $this->id );
        $this->thread_id = apply_filters(
'better_messages_message_thread_id_before_save', $this->thread_id, $this->id );
        $this->subject   = apply_filters(
'better_messages_message_subject_before_save', $this->subject, $this->id );
        $this->message   = apply_filters(
'better_messages_message_content_before_save', $this->message, $this->id );
        $this->date_sent = apply_filters(
'better_messages_message_date_sent_before_save', $this->date_sent, $this->id );
        $this->created_at = apply_filters(
'better_messages_message_created_at_before_save', $this->created_at, $this->id );
        $this->updated_at = apply_filters(
'better_messages_message_updated_at_before_save', $this->updated_at, $this->id );
        $this->temp_id = apply_filters(
'better_messages_message_temp_id_before_save', $this->temp_id, $this->id );

        /**
         * Fires before the current message item gets saved.
         *
         * Please use this hook to filter the properties above. Each part will be
passed in.
         */

```

```

* @since 1.0.0
*
* @param BP_Messages_Message $this Current instance of the
message item being saved. Passed by reference.
*/
do_action_ref_array( 'better_messages_message_before_save', array(
&$this ) );

// Make sure we have at least one recipient before sending.
if ( empty( $this->recipients ) ) {
    return false;
}

if( ! $wpdb->has_cap( 'utf8mb4' ) ){
    $this->subject = wp_encode_emoji( $this->subject );
    $this->message = wp_encode_emoji( $this->message );
}

$new_thread = false;

// If we have no thread_id then this is the first message of a new thread.
if ( empty( $this->thread_id ) ) {
    if( ! $wpdb->insert( bm_get_table('threads'), [ 'subject' => $this-
>subject ], [ '%s' ] ) ){
        return false;
    }

    $this->thread_id = $wpdb->insert_id;
    $new_thread = true;
}

// First insert the message into the messages table.
if ( ! $wpdb->query( $wpdb->prepare( "INSERT INTO " .
bm_get_table('messages') . " ( thread_id, sender_id, message, date_sent, created_at,
updated_at, temp_id ) VALUES ( %d, %d, %s, %s, %d, %d, %s )", $this->thread_id,
$this->sender_id, $this->message, $this->date_sent, $this->created_at, $this-
>updated_at, $this->temp_id ) ) ) {
    return false;
}

$this->id = $wpdb->insert_id;

$recipient_ids = array();

```

```

$time = Better_Messages()->functions->get_microtime();

if ( $new_thread ) {
    $unread_count = ( $this->count_unread ) ? 1 : 0;
    // Add an recipient entry for all recipients.
    foreach ( (array) $this->recipients as $recipient ) {
        $wpdb->query( $wpdb->prepare( "
        INSERT INTO
        " . bm_get_table('recipients') . "
        ( user_id, thread_id, unread_count, last_update )
        VALUES ( %d, %d, %d, %d )", $recipient->user_id, $this-
>thread_id, $unread_count, $time ) );
        $recipient_ids[] = $recipient->user_id;
    }

    // Add a sender recipient entry if the sender is not in the list of
recipients.
    if ( ! in_array( $this->sender_id, $recipient_ids ) ) {
        $wpdb->query( $wpdb->prepare(
        "INSERT INTO " . bm_get_table('recipients') . "
        ( user_id, thread_id, last_update )
        VALUES ( %d, %d, %d )", $this->sender_id, $this->thread_id,
$time ) );
    }
} else {
    if( $this->count_unread ) {
        $wpdb->query(
        $wpdb->prepare(
        "UPDATE " . bm_get_table('recipients') . "
        SET unread_count = unread_count + 1, is_deleted = 0,
last_update = %d
        WHERE thread_id = %d AND user_id != %d",
        $time,
        $this->thread_id,
        $this->sender_id
        )
        );
    } else {
        $wpdb->query(
        $wpdb->prepare(
        "UPDATE " . bm_get_table('recipients') . "
        SET is_deleted = 0, last_update = %d
        WHERE thread_id = %d AND user_id != %d",

```

```

        $time,
        $this->thread_id,
        $this->sender_id
    )
    );
}
}

//wp_cache_delete('thread_' . $this->thread_id, 'bm_messages');
//Better_Messages()->hooks->clean_thread_cache( $this->thread_id );

/**
 * Fires after the current message item has been saved.
 *
 * @since 1.0.0
 *
 * @param BP_Messages_Message $this Current instance of the
message item being saved. Passed by reference.
 */
do_action_ref_array( 'better_messages_message_after_save', array(
&$this ) );

return $this->id;
}

/**
 * Get a list of recipients for a message.
 *
 * @return object $value List of recipients for a message.
 */
public function get_recipients() {
    global $wpdb;

    return $wpdb->get_results( $wpdb->prepare( "SELECT user_id FROM
" . bm_get_table('recipients') . " WHERE thread_id = %d", $this->thread_id ) );
}

/** Static Functions
*****/

/**
 * Get list of recipient IDs from their usernames.
 *
 * @param array $recipient_usernames Usernames of recipients.

```

```

*
* @return bool|array $recipient_ids Array of Receptient IDs.
*/
public static function get_recipient_ids( $recipient_usernames ) {
    $recipient_ids = false;

    if ( ! $recipient_usernames ) {
        return $recipient_ids;
    }

    if ( is_array( $recipient_usernames ) ) {
        $rec_un_count = count( $recipient_usernames );

        for ( $i = 0, $count = $rec_un_count; $i < $count; ++ $i ) {
            if ( $rid = bp_core_get_userid( trim( $recipient_usernames[ $i ] ) ) )
        ) {
                $recipient_ids[] = $rid;
            }
        }
    }

    /**
     * Filters the array of recipients IDs.
     *
     * @since 2.8.0
     *
     * @param array $recipient_ids Array of recipients IDs that were
retrieved based on submitted usernames.
     * @param array $recipient_usernames Array of recipients usernames
that were submitted by a user.
     */
    return apply_filters( 'better_messages_message_get_recipient_ids',
$recipient_ids, $recipient_usernames );
}

/**
thread.
 * Get the ID of the message last sent by the logged-in user for a given
 *
 * @param int $thread_id ID of the thread.
 *
 * @return int|null ID of the message if found, otherwise null.
 */
public static function get_last_sent_for_user( $thread_id ) {

```

```

        global $wpdb;

        $query = $wpdb->get_var( $wpdb->prepare( "SELECT id FROM " .
bm_get_table('messages') . " WHERE sender_id = %d AND thread_id = %d ORDER
BY created_at DESC LIMIT 1", Better_Messages()->functions-
>get_current_user_id(), $thread_id ) );

        return is_numeric( $query ) ? (int) $query : $query;
    }

    /**
     * Check whether a user is the sender of a message.
     *
     * @param int $user_id ID of the user.
     * @param int $message_id ID of the message.
     *
     * @return int|null Returns the ID of the message if the user is the
     *         sender, otherwise null.
     */
    public static function is_user_sender( $user_id, $message_id ) {
        global $wpdb;

        $query = $wpdb->get_var( $wpdb->prepare( "SELECT id FROM " .
bm_get_table('messages') . " WHERE sender_id = %d AND id = %d", $user_id,
$message_id ) );

        return is_numeric( $query ) ? (int) $query : $query;
    }

    /**
     * Get the ID of the sender of a message.
     *
     * @param int $message_id ID of the message.
     *
     * @return int|null The ID of the sender if found, otherwise null.
     */
    public static function get_message_sender( $message_id ) {
        global $wpdb;

        $query = $wpdb->get_var( $wpdb->prepare( "SELECT sender_id
FROM " . bm_get_table('messages') . " WHERE id = %d", $message_id ) );

        return is_numeric( $query ) ? (int) $query : $query;
    }

```

Додаток Г – Ілюстративний матеріал

ІЛЮСТРАТИВНА ЧАСТИНА

**РОЗРОБКА ІНТЕРАКТИВНОЇ КОМУНІКАЦІЙНОЇ СИСТЕМИ
ПЕРЕДАЧІ ПОВІДОМЛЕНЬ**

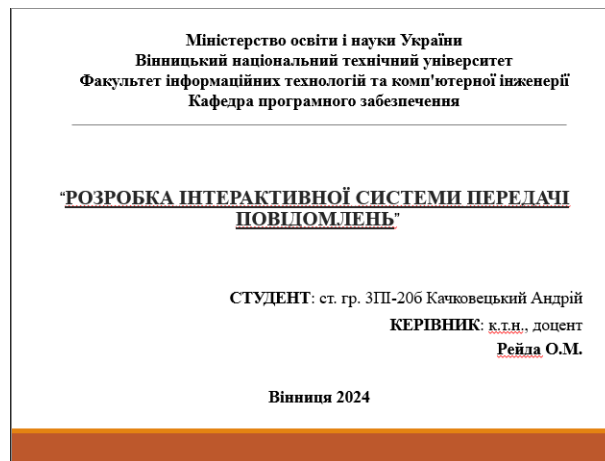


Рисунок Г.1 – Титульний слайд

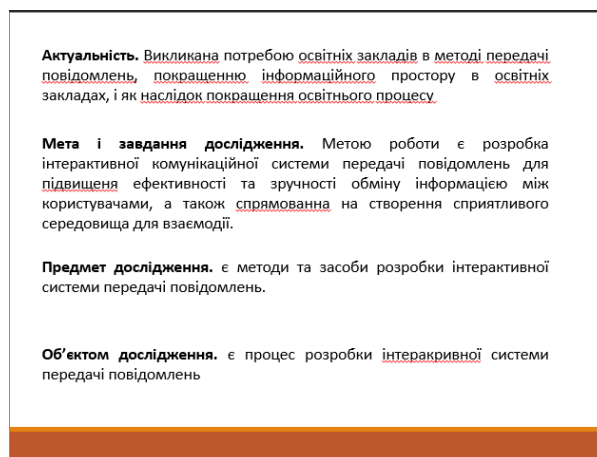


Рисунок Г.2 – Слайд «Актуальність, Мета і Завдання, предмет та область дослідження»

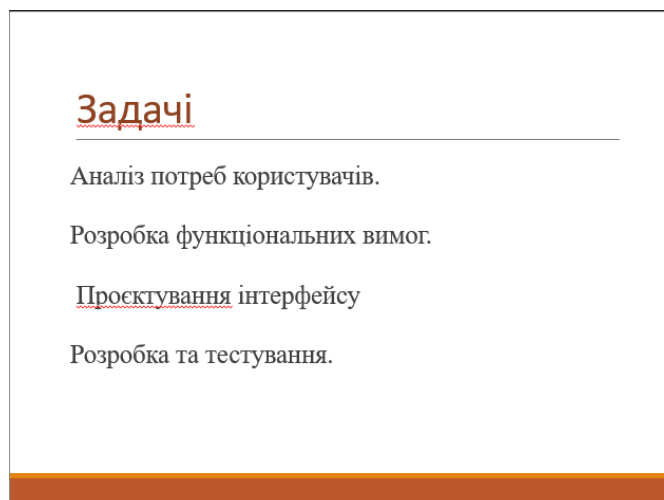


Рисунок Г.3 – Слайд «Задачі»

Наукова новизна

Подальшого розвитку отримав метод аналізу настроїв та емоційного стану користувачів, у якому, на відміну від існуючих, використано штучний інтелект для аналізу емоційного тону повідомлень, що дозволило виявляти настрої користувачів, відповідним чином реагувати, і, як наслідок, зменшити емоційне навантаження при спілкуванні.

Рисунок Г.4 – Слайд «Наукова новизна»

Практичне значення отриманих результатів

Практична цінність одержаних результатів полягає в тому, що на основі отриманих в бакалаврській дипломній роботі теоретичних положеннях було запропоновано сучасні підходи до розробки інтерактивних систем передачі повідомлень, розроблено програмні модулі передачі повідомлень, аналізу емоційного наповнення повідомлень.

Рисунок Г.5 – Слайд «Практичне значення результатів»

Аналоги

qq



Рисунок Г.6 – Слайд «Аналог QQ»

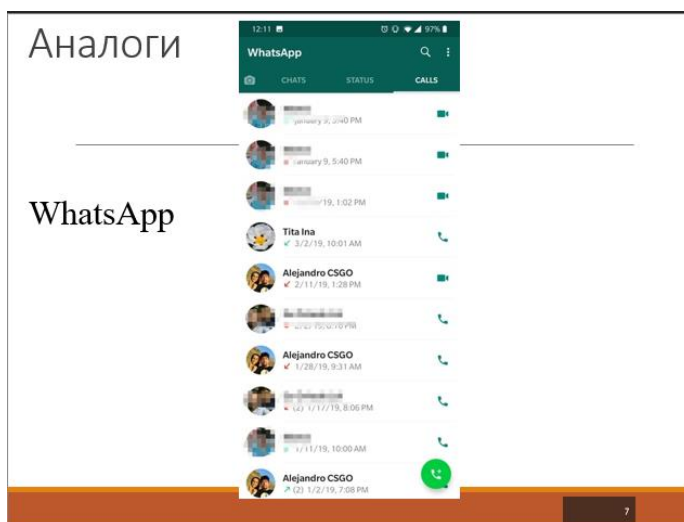


Рисунок Г.7 – Слайд «Аналог WhatsApp»

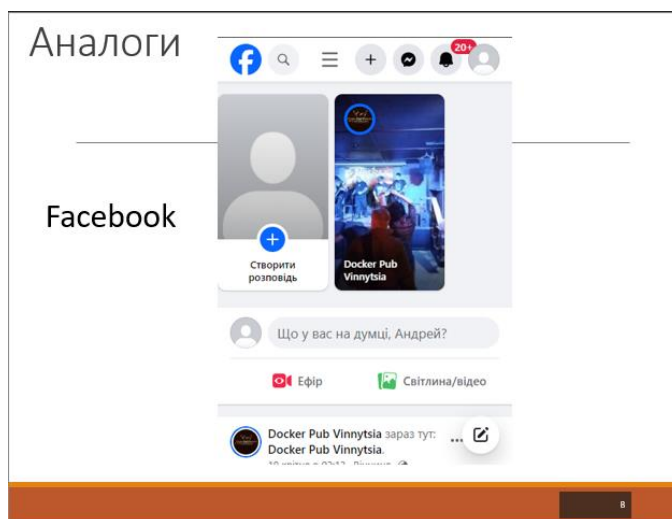


Рисунок Г.8 – Слайд «Аналог Facebook»

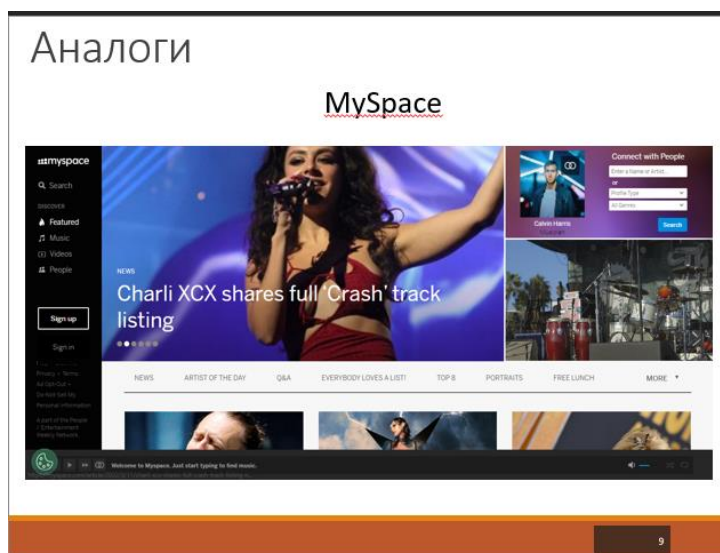


Рисунок Г.9 – Слайд «Аналог Myspace»

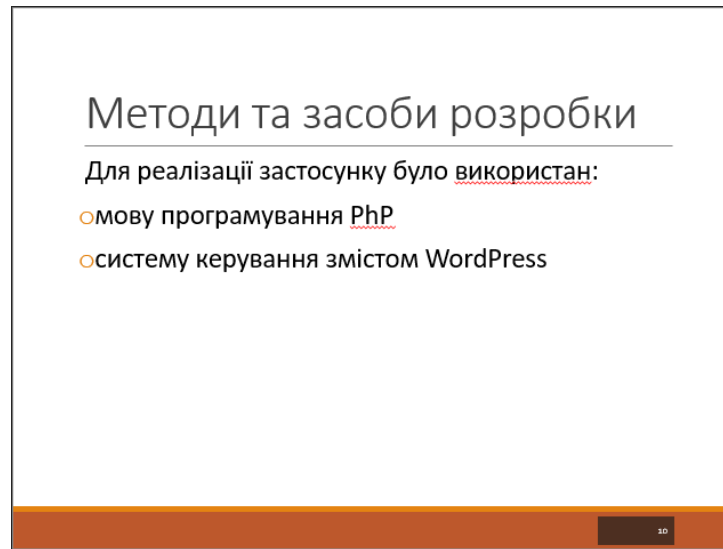


Рисунок Г.10 – Слайд «Методи та засоби розробки»

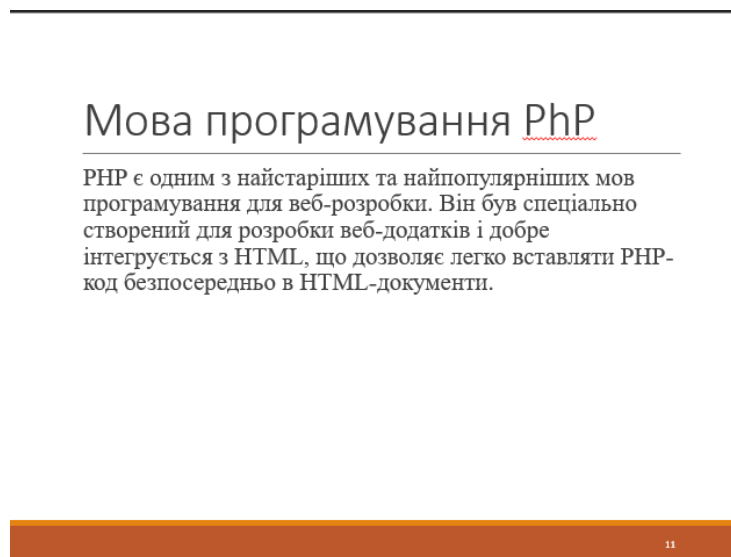


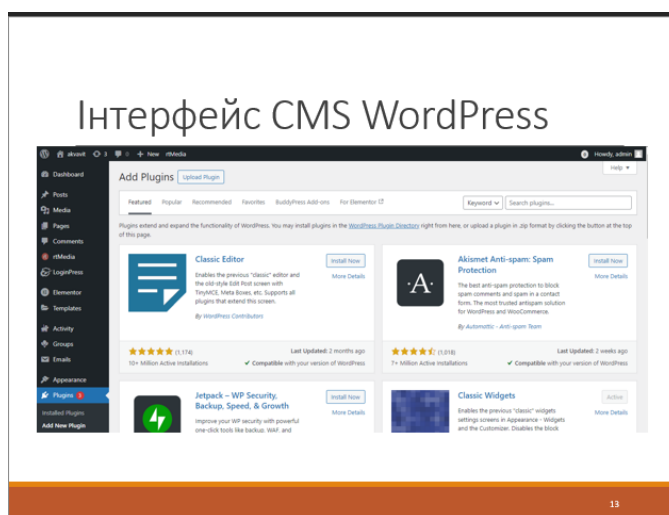
Рисунок Г.11 – Слайд «PHP»

Система керування змістом WordPress

WordPress є найбільш популярною CMS у світі, на яку припадає понад 40% всіх веб-сайтів. Його успіх зумовлений кількома ключовими факторами. По-перше, WordPress має надзвичайно зручний інтерфейс, що робить його доступним для новачків. Навіть користувачі без технічного досвіду можуть легко створювати та управляти веб-сайтами завдяки інтуїтивно зрозумілій панелі адміністрування. По-друге, WordPress пропонує велику кількість безкоштовних та платних тем і плагінів, що дозволяє розширювати функціональність сайту без необхідності програмування. Це робить його надзвичайно гнучким та адаптивним до різних потреб.

12

Рисунок Г.12 – Слайд «WordPress»



13

Рисунок Г.13 – Слайд «Інтерфейс WordPress»



Рисунок Г.14 – Слайд «Структура застосунку»

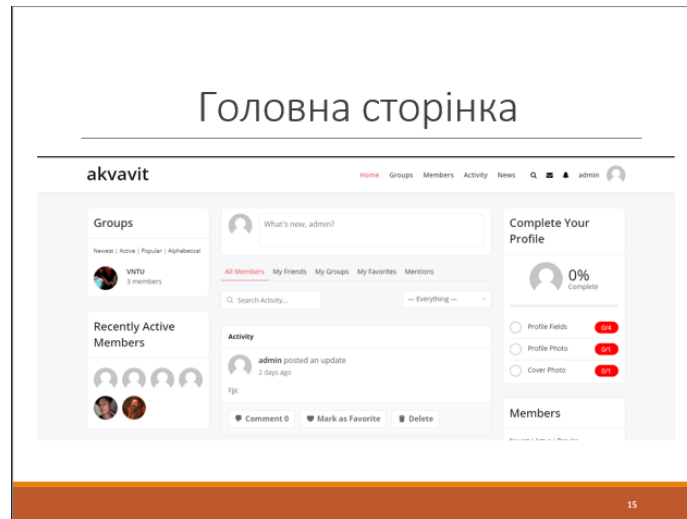


Рисунок Г.15 – Слайд «Головна сторінка»

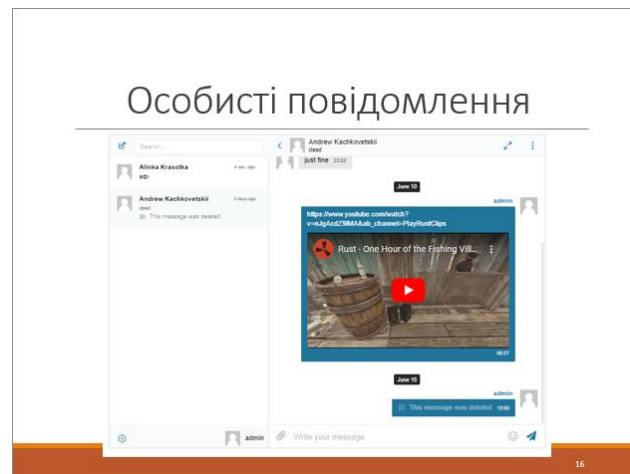


Рисунок Г.16 – Слайд «Особисті повідомлення»

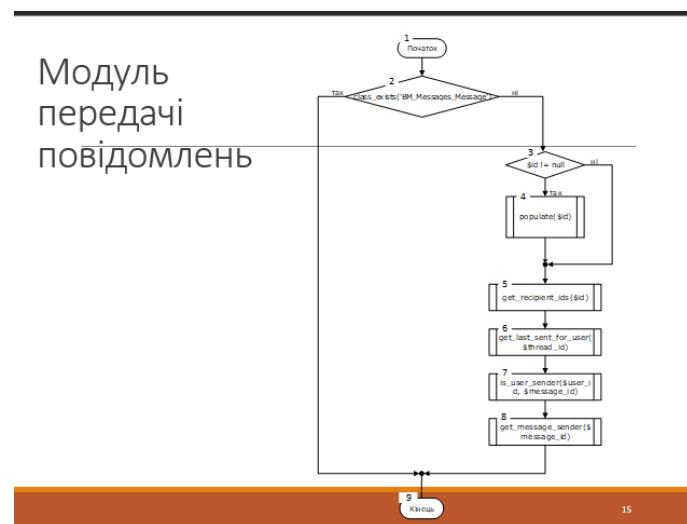


Рисунок Г.17 – Слайд «Модуль передачі повідомлень»

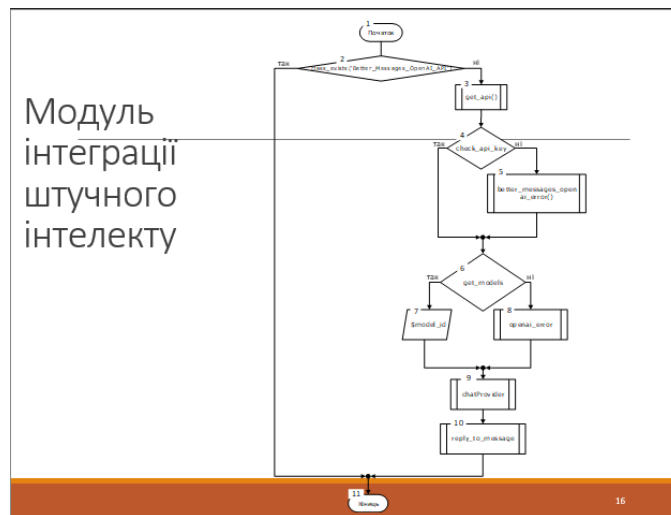


Рисунок Г.18 – Слайд «Модуль Інтеграції ШІ»

Висновки

- Проведено аналіз існуючих рішень систем передачі повідомлень
- Було розроблено інтерфейс додатку
- Проведено варіантний аналіз і обґрунтування вибору методів та засобів для реалізації програмного забезпечення
- Розроблено модуль передачі повідомлень
- Розроблено модуль інтегрування штучного інтелекту
- Проведено тестування та оцінку ефективності розробленої системи в реальних умовах

Рисунок Г.19 – Слайд «Висновки»

Публікації й апробації

УДК 331.108

КАЧКОВЕЦЬКИЙ А.І.
РЕЙДА О.М.

ПЕРСПЕКТИВИ ІНТЕРАКТИВНИХ СИСТЕМ ПЕРЕДАЧІ ПОВІДОМЛЕНЬ

Національний національний технічний університет

Анотація
Робота «Перспективи Інтерактивних Систем Передачі Повідомлень» присвячена розгляду перспектив, перспективних перспектив інтеграції інтеграції системи передачі повідомлень. Аналізуються результати аналізу систем передачі повідомлень та здійснюється вибір методів та засобів для реалізації програмного забезпечення. Робота також висвітлює можливість реалізації системи передачі повідомлень в реальних умовах. Робота висвітлює методи, які можуть привести до подальшого розвитку та розвитку інтеграції системи передачі повідомлень в інтеграції.

Ключові слова: перспективи, інтерактивні системи, розвиток.

Abstract
The work "Prospects of Interactive Message Transmission Systems" is devoted to the consideration of prospects, the use of various applications of integrated message transmission systems. Attention is focused on the analysis of key trends and future directions of development in this field. The work also highlights the possibilities of implementing technology in the fields of interactive, objective and accurate to people with disabilities. In general, the work that can lead to further growth and development of interactive message transmission systems in the future are analyzed.

Keywords: perspectives, integration systems, development.

Вступ
Інтерактивні системи передачі повідомлень (ІСПП) є важливою частиною сучасної комунікації. З розвитком технологій ці системи мають значний вплив на спосіб обміну інформацією між людьми. В даному дослідженні розглядаються перспективи розвитку ІСПП, їх роль у сучасному світі, а також потенційні виклики та можливості.
На сьогодні ІСПП є надзвичайно популярними і мають широкий спектр застосувань, починаючи від особистого спілкування до професійного використання в бізнесі. Вони забезпечують швидкість, швидкість та ефективність комунікації.

Результати досліджень
При розгляді перспектив у використанні програмних засобів інтерактивних систем передачі повідомлень, слід звернути увагу на наступні аспекти:

Рисунок Г.20 – Слайд «Апробації та публікація результатів роботи»

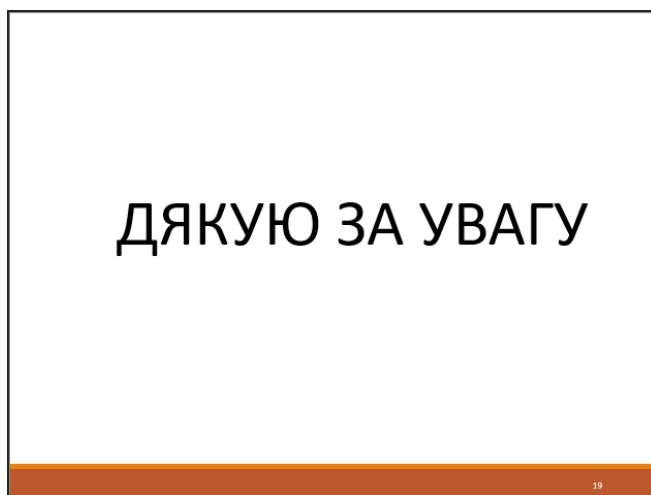


Рисунок Г.21 – Фінальний слайд