

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
**Центр заочної, дистанційної та вечірньої форм навчання**  
**Кафедра інформаційних технологій**

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня магістр**

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Web-орієнтована система консультування ІТ-фахівців з професійних питань

Здобувачки групи ІТ.мз-21с Головенко Вікторії Вікторівни  
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_  
(підпис)

Вікторія ГОЛОВЕНКО  
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник \_\_\_\_\_ к.т.н., доц., Юлія ПАРФЕНЕНКО \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Сумський державний університет  
Центр заочної, дистанційної та вечірньої форм навчання  
Кафедра інформаційних технологій  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

## **ЗАВДАННЯ**

**на кваліфікаційну роботу магістра студентіві**

Головенко Вікторія Вікторівна

(прізвище, ім'я, по батькові)

**1 Тема кваліфікаційної роботи** Web-орієнтована система консультування ІТ-фахівців з професійних питань

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

**2 Термін здачі студентом кваліфікаційної роботи** «10» грудня 2023 р.

**3 Вхідні дані до кваліфікаційної роботи** інформація про консультації, інформація про користувачів, запит користувача, інформація про штучний інтелект

**4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** вступ, аналіз предметної області, огляд останніх досліджень і публікацій, аналіз сервісів-аналогів систем консультування, постановка задачі та методи дослідження, мета та задачі дослідження, розробка візуальної частини web-орієнтованої системи консультування, розробка функціональної частини web-орієнтованої системи консультування, моделювання та проектування, структура web-орієнтованої системи консультування, структурно-функціональне моделювання процесу, моделювання діаграми варіантів використання web-орієнтованої системи консультування, проектування бази даних, практична реалізація web-орієнтованої системи консультування, програмна реалізація, використання web-орієнтованої системи користувач, адміністрування web-орієнтованої системи, тестування та дослідження працездатності web-орієнтованої системи, висновки, список використаних джерел.

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** контекстна діаграма web-орієнтованої системи по процесу надання відповіді на запитання, декомпозиція IDEF0 діаграми web-орієнтованої системи по процесу

надання відповіді на запитання, діаграми варіантів використання, пакети та модулі web-орієнтованої системи консультування

**6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання \_\_\_\_\_.

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів проекту	Примітка
1	Аналітика предметної області	04.09.23-11.09.23	
2	Створення технічного завдання	11.09.23-26.09.23	
3	Моделювання та проектування	27.09.23-16.10.23	
4	Розробка бази даних	16.10.23-26.10.23	
5	Практична реалізація web-орієнтованої системи	26.10.23-17.11.23	
6	Тестування та наповнення контентом	17.11.23-26.11.23	
7	Формування пояснювальної записки	26.11.23-27.12.23	

Магістрант \_\_\_\_\_

Головенко В.В.

Керівник роботи \_\_\_\_\_

к.т.н., доц. Парфененко Ю.В.

## АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Web-орієнтована система консультування IT-фахівців з професійних питань».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 41 найменувань, 3 додатків. Загальний обсяг роботи – 111 сторінок, у тому числі 77 сторінок основного тексту, 4 сторінки списку використаних джерел, 30 сторінок додатків.

Актуальність роботи полягає в використанні генеративного штучного інтелекту як складової частини інформаційного середовища для покращення взаємодії між фахівцем і користувачем у різних аспектах.

Мета роботи: розробити web-орієнтовану систему консультування IT-фахівців у сфері професійних питань.

При виконанні кваліфікаційної роботи було вивчено сферу консультацій та проведено аналіз актуальних наукових досліджень у цьому напрямку. Також було розглянуто публікації та сучасні конкуруючі продукти в цій області.

Крім того, було визначено основну мету створення web-орієнтованої системи консультування. Після проведеного аналізу були вибрані програмні засоби та інструментарій, які найкраще відповідають цьому завданню.

Для подальшої розробки було виконано проектування web-орієнтованої системи, проведено детальне описання процесу через структурно-функціональне моделювання та розглянуто сценарії використання.

Також було розглянуто приклади використання web-орієнтованої системи консультування для IT-фахівців щодо професійних питань з боку як користувача та адміністратора. Всі можливості системи були відображені та детально протестовані.

Ключові слова: web-орієнтована система, консультування, фахівець, питання, штучний інтелект, форум, рейтинг, користувач, адміністратор.

## ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень і публікацій.....	8
1.2 Аналіз сервісів-аналогів систем консультування.....	13
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	24
2.1. Мета та задачі дослідження.....	24
2.2. Розробка візуальної частини web-орієнтованої системи консультування.....	25
2.3. Розробка функціональної частини web-орієнтованої системи консультування.....	28
3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ.....	30
3.1 Структура web-орієнтованої системи консультування.....	30
3.2 Структурно-функціональне моделювання.....	31
3.3 Моделювання діаграми варіантів web-орієнтованої системи консультування.....	34
3.4. Проектування бази даних.....	38
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ КОНСУЛЬТУВАННЯ.....	42
4.1 Програмна реалізація.....	42
4.2 Використання web-орієнтованої системи користувачем.....	43
4.3 Адміністрування web-орієнтованої системи.....	63
4.4 Тестування та дослідження працездатності web-орієнтованої системи.....	65
ВИСНОВКИ.....	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	78
ДОДАТОК А.....	82
ДОДАТОК Б.....	95
ДОДАТОК В.....	101

## ВСТУП

Почала активно зростати проблема отримання вичерпних відповідей на конкретні запитання та роз'яснень від фахівців у різних сферах галузі. Зустрічі, які колись проводилися в реальному житті, стали інтерактивним обміном інформацією в онлайн-середовищі.

Зростає потреба у консультативній допомозі у форматі онлайн у сфері інформаційного обміну, і це може викликати накопичення проблем. Користувачам важко знайти вільного фахівця з відповідними знаннями. Адже вони не завжди можуть оперативно надати консультації. Саме тому слід придивитися в бік, набираючого популярності, штучного інтелекту.

Для покращення оперативної відповіді на цілком конкретні та прості запитання слід використати штучний інтелект. Допомога штучного інтелекту користувачеві в пошуку відповідей на питання – корисний та доступний функціонал. Основний сенс цієї допомоги полягає в тому, щоб забезпечити користувачам швидкий та ефективний доступ до інформації, яка їм потрібна. Особлива увага приділяється діалогу між штучним інтелектом та користувачем.

Для успішної реалізації web-орієнтованої системи потрібно зосередити увагу на всіх важливих аспектах, щоб основні функції були завжди легкодоступними та інтуїтивно зрозумілими для користувачів.

**Об'єкт дослідження:** процес інформаційної підтримки ІТ-фахівців при консультуванні з професійних питань.

**Предмет дослідження:** інформаційне забезпечення консультування ІТ-фахівців з професійних питань.

**Мета:** розробити web-орієнтовану систему консультування ІТ-фахівців у сфері професійних питань.

Особливістю запропонованої системи буде використання генеративного штучного інтелекту як складової частини інформаційного середовища для покращення взаємодії між фахівцем і користувачем у різних аспектах.

**Задачі:**

- ознайомлення з предметною областю інформаційного забезпечення консультування ІТ фахівців;
- аналіз функціональних можливостей web-орієнтованих систем у сфері консультування;
- розробка вимог до проєкт web-орієнтованої системи;
- прототипування web-орієнтованої системи консультування;
- структурно-функціональне моделювання процесу консультування ІТ фахівців з використанням веборієнтований інформаційної системи;
- вибір інструментів та бібліотек для розробки системи;
- розробка візуальної та функціональної частини системи
- завантаження web-орієнтованої системи консультування на сервер;
- тестування системи консультування.

Практичне значення розробки web-орієнтованої системи для ІТ-фахівців у сфері професійних питань, яка використовує штучний інтелект для пришвидшення пошуку правильних відповідей в різних темах інформаційних технологій, полягає у прискорення пошуку потрібної відповіді користувачеві для спрощення консультації в режимі онлайн в межах інформаційної системи.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень і публікацій

Консультавання в інформаційному просторі має розгалуження за видами реалізації представлення інформації, способом її подачі та комфортом використання того чи іншого сервісу.

Форуми та групи обговорень є популярним місцем, де люди можуть обмінюватися досвідом та питаннями щодо різних аспектів ІТ. Це включає загальні теми, такі як програмування, мережі, безпека та конкретні платформи або мови програмування [1].

Форуми поділяються на різні тематичні розділи, де користувачі можуть обирати область, яка відповідає їхнім інтересам або питанням. Наприклад, можуть бути розділи, присвячені програмуванню, мережам, адмініструванню систем, розробці web-сайтів тощо. Користувачі можуть створювати нові теми для обговорень або долучатися до існуючих. Вони можуть поставити питання, викладати проблеми, поділитися новими відкриттями та рішеннями.

Форуми, як онлайн-платформи зазвичай залежать від спільноти користувачів. Якщо на форумі є досвідчені фахівці, вони можуть відповідати на питання та надавати консультації [2]. Це може бути особливо корисно для новачків, які шукають допомогу, але і робить сильну залежність від наявності досвідчених людей в тій чи іншій області для оперативного реагування на появу нових питань.

Деякі форуми дозволяють користувачам залишати повідомлення анонімно або під псевдонімами, що дозволяє більш вільно обговорювати чутливі або особисті питання [3].

Інформаційні ресурси у вигляді блогів пропонують інформацію та поради щодо різних аспектів ІТ. Цей вид консультавання надає доступ до статей, публікацій, порад та інших інформаційних ресурсів, які допомагають ІТ-спеціалістам оновлювати свої навички та дізнаватися про останні новини та технологічні тенденції в галузі ІТ.



Прикладом однією із таких систем є компанія L&Tsoftware technologies. Зі статті [4] можна дізнатися, що вебсайт було розроблено на основі Mern Stack. Головною цілю розробки сайту є використання для пошуку продуктів і послуг, що надаються консультантом, без будь-якого фізичного відвідування.

Багато блогів та інформаційних ресурсів публікують статті та публікації на різноманітні теми в галузі ІТ, включаючи розробку програмного забезпечення, адміністрування мереж, кібербезпеку, інтернет-маркетинг, аналіз даних та багато інших. Ці матеріали можуть містити важливу інформацію та рекомендації для ІТ-спеціалістів. Саме статтями та публікаціями блоги вважаються дуже комфортними та приваблюють все більше користувачів [5].

Блогери та автори інформаційних ресурсів часто діляться своїм досвідом, надаючи поради та рекомендації щодо різних аспектів роботи. Користувачі можуть залишати коментарі та відгуки, що допомагає іншим отримати додаткову інформацію та розуміння. Крім того, це не обмежується ІТ сферою, гарним прикладом є робота в індустрії онлайн навчання [6]. У даній роботі описується впровадження інструментів Web 2.0 для вдосконалення інструментів онлайн консультування. Всі розробки в організаціях були виконані, щоб визначити загальні проблеми, які необхідно вирішити. Особливий акцент був зроблений на допомогу організації роботи з літератури.

Блоги та інформаційні ресурси можуть бути важливим джерелом останніх новин та оновлень в ІТ-галузі. ІТ-спеціалісти можуть слідкувати за новими технологіями, змінами в програмному забезпеченні та іншими подіями в галузі завдяки цим ресурсам [7]. Деякі блоги та інформаційні ресурси мають розділи для дискусій та обговорення, де читачі можуть задавати питання, обмінюватися думками та досвідом. Це сприяє активній взаємодії між ІТ-спеціалістами.

Блоги уніфікують даний вид консультацій та дозволяє ІТ-професіоналам розвивати свої навички та здобувати нові знання в зручний спосіб [8]. Але подібні інформаційні ресурси можуть містити не лише аналізи та рецензії на різні ІТ-продукти, програмне забезпечення та технології, а й вигадані факти, із-за чого

подібний вид отримання швидкої відповіді ставить під сумнів актуальність наданої інформації.

Загалом блоги та форуми є головними інструментами для ІТ-спеціалістів, щоб не лише отримати швидко відповідь на питання, а й залишатися в курсі останніх подій та розвивати свої професійні навички.

Система консультування за підтримки штучного інтелекту може не лише уніфікувати та прискорити пошук правильних відповідей, а й має додаткові переваги. Використовуючи потужні алгоритми пошуку та індексації, можна швидко знаходити відповіді на запити. Штучний інтелект може сканувати та аналізувати величезну кількість інформації, щоб вивести точний результат [9].

Штучний інтелект може надати швидкі та автоматизовані відповіді на загальні або типові запитання ІТ-спеціалістів. Це допомагає значно скоротити час очікування відповіді. Аналіз великих обсягів даних дозволяє штучному інтелекту надати рекомендації та підказки ІТ-спеціалістам [10]. Наприклад, він може виявляти аномалії у системах, що вказують на можливі проблеми, або аналізувати дані про використання ресурсів для оптимізації процесів.

Можливість навчання і постійного вдосконалення важко недооцінити та показує, що штучний інтелект може постійно вдосконалювати свої алгоритми та здатність робити більш точні прогнози та рекомендації. Одна із головних переваг штучного інтелекту – обробка запитів у будь-який час оскільки може працювати без перерви на відпочинок. Це означає, що користувачі можуть отримувати допомогу та відповіді на свої запитання навіть під час нічного часу або у вихідні. Це в свою чергу веде до зниження навантаження на ІТ-персонал який консультує користувачів та звільняє ІТ-спеціалістів від монотонних обов'язків з рутинними питаннями і дозволяє їм зосередитися на складніших завданнях [11].

Покращення ефективності і якості обслуговування клієнтів: штучний інтелект може надати точні та послідовні відповіді на запитання, знижуючи ймовірність помилок та неправильних рішень. Це допомагає підвищити якість обслуговування ІТ-клієнтів.

Розширення географії консультацій: штучний інтелект може надавати консультації віддалено, що дозволяє ІТ-спеціалістам працювати з клієнтами з будь-якої точки світу.

В інформаційній системі штучний інтелект може виконувати роль асистента. Цей асистент здатен відповідати на запитання, розв'язувати технічні проблеми та надавати консультації з різних аспектів ІТ [12].

ІТ-спеціаліст може ввести своє запитання або описати проблему, з якою він стикається. Інтелектуальний асистент аналізує запитання та шукає відповіді в базі знань. Він також може аналізувати подібні запитання, які раніше вирішувалися, і надавати рекомендації.

Штучний інтелект – гнучкий інструмент, який може допомогти також в написанні коду. Якщо ІТ-спеціаліст має проблему з кодом або програмуванням, він може надіслати фрагмент коду асистенту. Штучний інтелект аналізує код, виявляє можливі помилки, пропонує виправлення та пояснює, як вони працюють [13].

Штучний інтелект може допомагати в прийнятті рішення. Якщо ІТ-спеціаліст стикається з складною проблемою, штучний інтелект може аналізувати альтернативи та надавати рекомендації для прийняття рішення. Також штучний інтелект може прискорити навчання нових навичок, вибираючи оптимальний шлях вивчення, рекомендуючи онлайн-курси, літературу та матеріали [14].

За допомогою штучного інтелекту ІТ-спеціалісти можуть отримувати надійну підтримку та консультації, розв'язувати технічні проблеми та покращувати свою продуктивність в галузі ІТ.

Для розробки систем консультування використовують Deep Learning [15]. Прикладом такої роботи є розробка системи чат-бот для консультацій в'єтнамських університетів. Система має три важливі завдання: розпізнавання намірів користувача, керування діалогом і канали відповідей. Для завдання розпізнавання намірів користувача метод зіставлення різних шаблонів.

Також для розробки типових систем консультування часто використовують Python як інструмент для розробки чат-ботів чи додаткових алгоритмів систем. Також

такі додатки можуть використовуватися у багатьох сферах. Розглянемо приклади вже готових систем.

Одним із прикладів такої роботи є веб-додаток дистанційного керування сільським господарством і консультування. Dodatok включає в себе також системи раннього діагностичного попередження. У цьому дослідженні [16] представлена інформація про веб-додаток дистанційного сільськогосподарського контролю та консультування, що був розроблений на мові програмування Python.

Консультування у медичній сфері не є виключення. Для вдосконалення вже існуючої системи консультування було використано Web Crawler та фрейм Scrapy на Python [17]. Система в основному реалізує сканування та зберігання даних про лікаря в певній області призначеному відділенні та відвідувачів.

В інших дослідженнях же було проаналізовано консультації пацієнтів через онлайн-спільноти охорони здоров'я. У дослідженнях [18-19] описується використання Python Spiders для отримання даних із [Naodf.com](http://Naodf.com) й роботу з більше ніж 2982 звітів лікарів. Крім того, за допомогою розробленої системи було проаналізовано аспекти соціальної підтримки лікарів та її вплив на консультації пацієнтів.

У тематиці юриспруденції ми можемо розглянути приклад із розробкою та дослідженням системи класифікації текстів юридичних консультацій [20]. Дослідження та розробку чат-боту було проведено на основі юридичної мережі Китаю, що включала в себе перелік основних запитів та типових відповідей.

Було розглянуто багато прикладів розробки та використання функціоналу на основі мови Python для консультування. Можна звернути увагу, що більшість із прикладів були чат-боти або окремі системи. Для розробки веб-додатків зазвичай використовуються інші мови програмування.

Розглянемо також приклади розробки систем по консультуванню на основі мови PHP. Першим прикладом є онлайн-консультаційна програма будівельної фірми. За статтею [21] можна зробити висновок, що застосування онлайн-консультаційної програми полегшила клієнтам отримання прямих консультацій, навіть якщо фахівець

перебуває за містом. Веб-додаток допомагає знайти відповіді на запитання без необхідності чекати та записуватися на зустріч з консультантом.

Не рідкістю є використання додатків для консультування у сферах навчання. Як наприклад, було проведено розробку інформаційної системи для консультацій у фінальному проекті на основі університету [22].

При розробці на РНР розробники використовують переваги того, що до веб-сайту можна отримати доступ, багато дій щодо проведення опитувань переносяться на багато веб-сайтів, які є динамічними [23]. Цією перевагою скористали й розробники веб-додатку для консультування в Бетаві.

## **1.2 Аналіз сервісів-аналогів систем консультування**

Аналіз аналогів є необхідним елементом стратегії розробки та маркетингу Web-орієнтованих систем і допомагає зробити їх більш конкурентоздатними і успішними на ринку.

Перший сервіс на розгляданні – Stack Overflow [24]. Важливий ресурс для програмістів та інших інформаційних технологій, проте, як і в усіх сервісах, він має як позитивні, так і негативні аспекти. Stack Overflow має потужну базу аудиторію, чим дуже сильно виділяється серед ІТ-спеціалістів.

Даний додаток використовується й в інших роботах [25], де пропонується спеціалізована архітектура глибокого навчання Post2Vec, яка витягує розподілені представлення повідомлень Stack Overflow і демонструє, що представлення можна використовувати для значного підвищення ефективності найсучасніших рішень для трьох завдань.

Представлення сервісу Stack Overflow на рисунку 1.1.

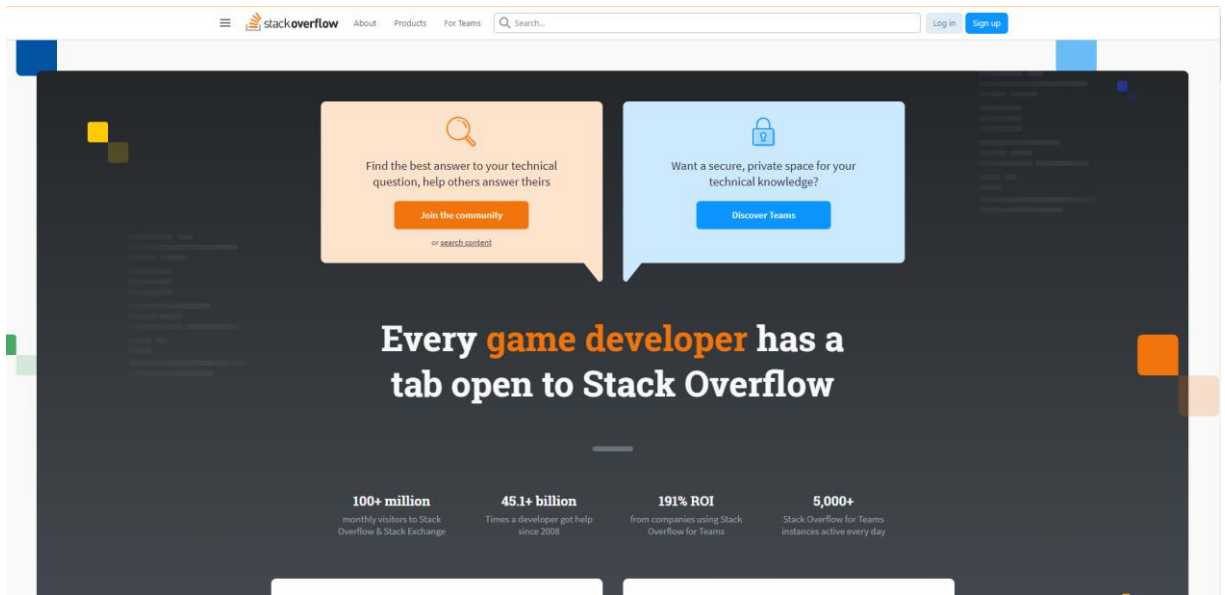


Рисунок 1.1 – Головна сторінка сайту «Stack Overflow»

*Джерело:* знімок з екрану [24]

Варто зазначити, що одразу знайти відповідь на потрібне запитання не вийде. Сервіс обов'язково попросить користувача зареєструватися або авторизуватися, що уповільнює пошук потрібної інформації. Але кольори приємні, не викають втомленості при довготривалому користуванні сервісом. Сторінка тем Stack Overflow на рисунку 1.2.

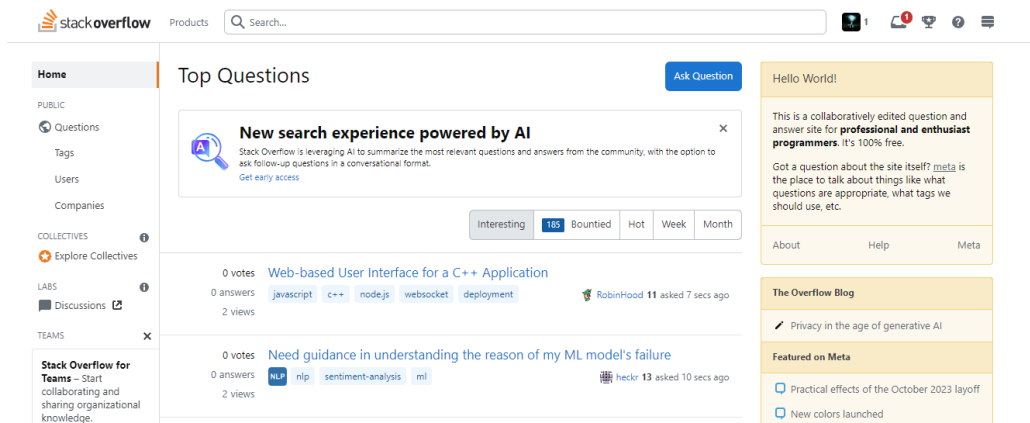


Рисунок 1.2 – Сторінка тем «Stack Overflow»

*Джерело:* знімок з екрану [24]

Загалом сервіс має інформативний інтерфейс, хоч і він не такий привабливий, щоб бути комфортним для нових вимогливих користувачів. Нажаль на будь-яку тему може відповісти кожен, тому якість відповідей ставиться під сумнів. Мінімалістичний інтерфейс показано на рисунку 1.3.

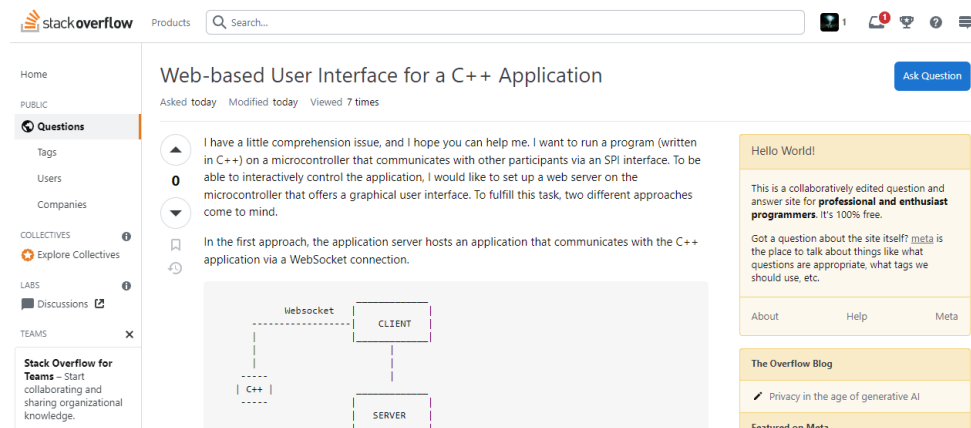


Рисунок 1.3 – Сторінка одної з тем «Stack Overflow»

*Джерело:* знімок з екрану [24]

В таблиці 1.1 представлені співставлені переваги та недоліки сервісу Stack Overflow.

Таблиця 1.1 – Порівняльна таблиця переваг та недоліків «Stack Overflow»

Переваги	Недоліки
Stack Overflow надає обширний обсяг корисної інформації та відповідей на питання, які можуть виникнути у програмістів. Це допомагає вирішувати проблеми та здобувати нові знання;	Оскільки будь-хто може давати відповіді на запитання, іноді можуть зустрічатися неточні або неправильні відповіді.
Спільнота Stack Overflow є великою та завжди активною, тому питання зазвичай отримують відповіді досить оперативно.	Багато запитань можуть бути схожими або навіть ідентичними, що може призвести до дублювання вмісту;

Продовження таблиці 1.1

Переваги	Недоліки
Користувачі можуть голосувати за якість відповідей, що допомагає виділити найкращі рішення для конкретних проблем;	Stack Overflow спеціалізується на технічних питаннях, іноді це важко сприймається новачками та тими, хто не має глибоких технічних знань;
Питання можуть бути позначені за певними темами або мовами програмування, що спрощує пошук відповідей;	Деякі публікації або обговорення можуть мати контент зі спірним вмістом;
Будь-який користувач може відповідати на питання та спільно вирішувати проблеми інших.	Іноді важко пояснити свою думку в спільноті, яка визначає, що є "правильним" чи "неправильним".

*Джерело:* побудовано автором

Загалом, Stack Overflow запам'ятався низьким рівнем якості відповідей, дублюванням тем та важкий для опанування новачками.

Наступний в порівнянні – Slashdot, який відомий як новинний та соціальний web-сайт, спеціалізований на інформації про технології та інтернет [26].

Експериментальні результати [27] на двох реальних наборах даних Slashdot і Epinions показують, що алгоритм D-RIS близький до алгоритму CELF і вище, ніж алгоритм RIS, алгоритм HighDegree, алгоритм LIR і алгоритм pVnH в діапазоні поширення впливу і значно кращий за алгоритм Celf і алгоритм RIS у часі виконання.

Демонстрація Slashdot на рисунку 1.4.



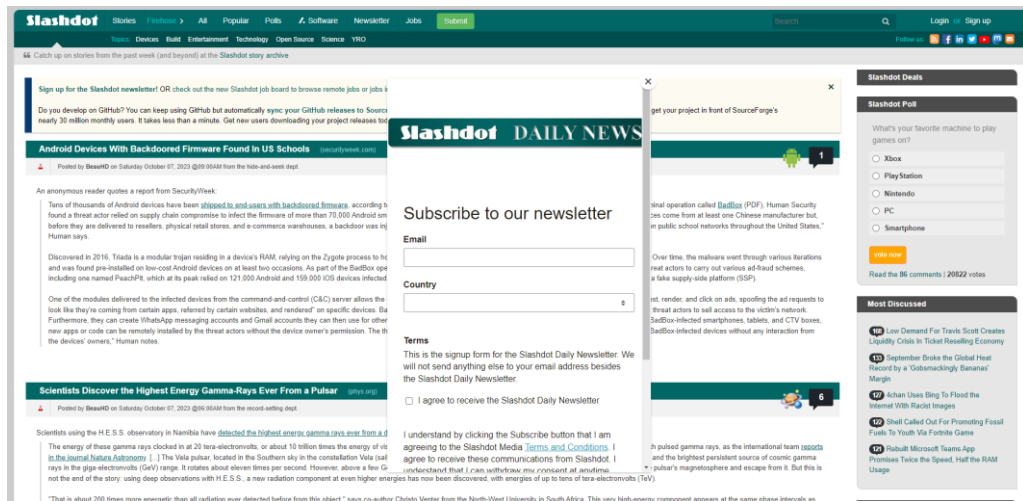


Рисунок 1.4 – Демонстрація роботи «Slashdot»

*Джерело: знімок з екрану [26]*

На жаль, точність рекомендацій часто навіть не пов'язана з темою, яку користувач переглядає, а пошукова система не видає бажаного результату. Демонстрація одного з записів на рисунку 1.5.

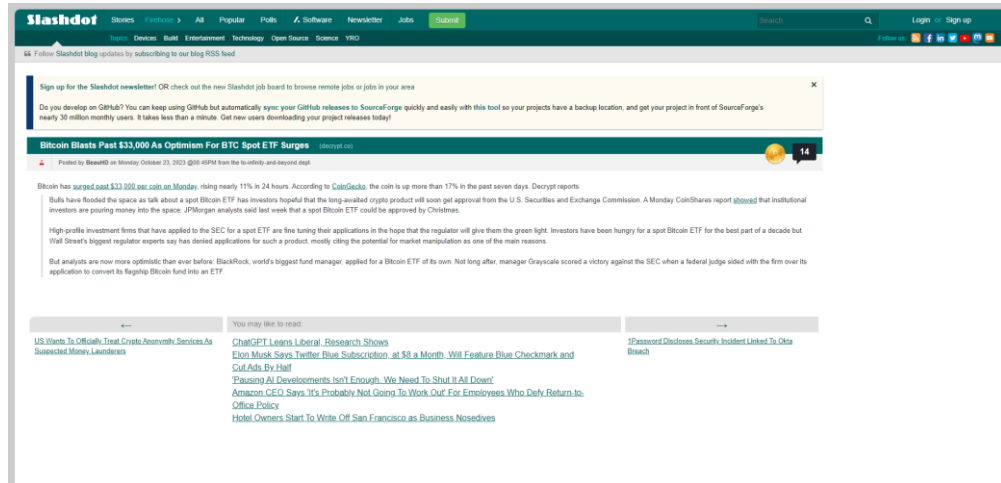


Рисунок 1.5 – Показ одного з записів «Slashdot»

*Джерело: знімок з екрану [26]*

Більшість тем майже не торкаються ІТ, а коментарі взагалі більше схоже на особисту переписку, ніж на обговорення теми, що показано на рисунку 1.6. Модерація під деякими темами взагалі відсутня. Дизайн та управління виглядає дуже застаріло та взагалі не приваблює користувачів.

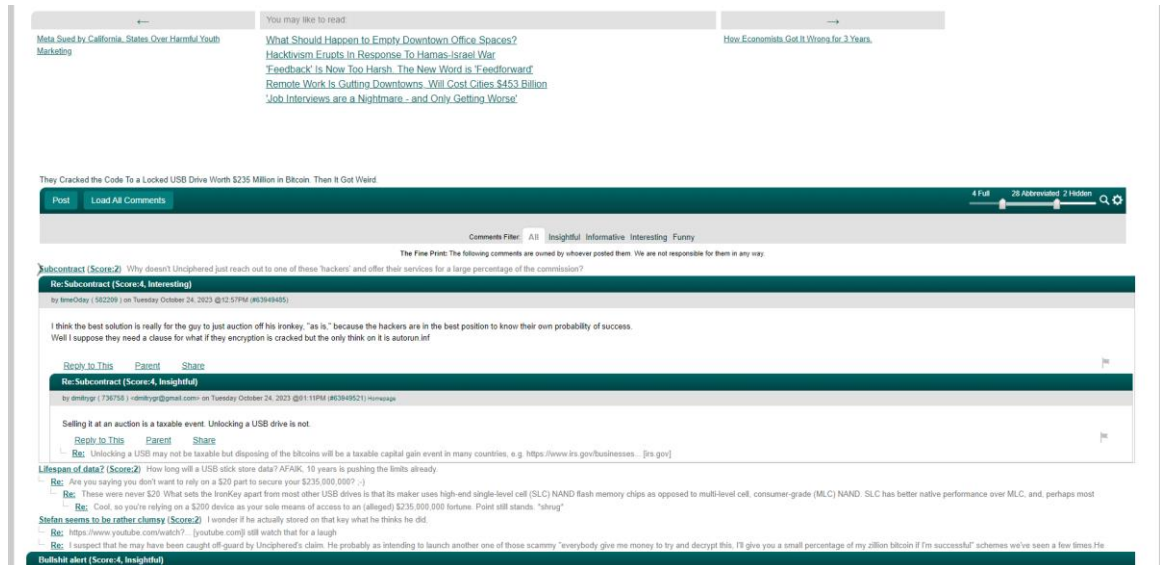


Рисунок 1.6 – Демонстрація коментарів «Slashdot»

Джерело: знімок з екрану [26]

В таблиці 1.2 представлені співставлені переваги та недоліки веб-додатку Slashdot.

Таблиця 1.2 – Порівняльна таблиця переваг та недоліків «Slashdot»

Переваги	Недоліки
Платформа Slashdot надає актуальні та цікаві новини з галузі технологій, інтернету, науки та інших суміжних областей, що особливо корисно для зацікавлених в інформаційних технологіях;	Оскільки Slashdot головним чином цікавиться технологіями, це може призвести до відчуття новими користувачами відчуженості серед інших користувачів;
Наявна велика та активна спільнота користувачів, яка не лише читає новини, а й обговорює їх, виражає власні думки та ділиться досвідом;	Застарілий дизайн та інтерфейс Slashdot виглядає застаріло в порівнянні зі сучасними стандартами;

## Продовження таблиці 1.2

Переваги	Недоліки
Для підвищення якості обговорень на платформі використовується система модерації коментарів, що допомагає виділити якісний вміст та зменшити кількість несуттєвих коментарів.	Система модерації призначена для поліпшення якості обговорень, але деякі модератори можуть вирішувати суб'єктивно, які коментарі вважати важливими або ховати;
	Slashdot обмежений галузями, пов'язаними з технологіями та наукою
	Обговорення на Slashdot часто мають технічну спрямованість та можуть вимагати глибоких знань у вибраній темі.

*Джерело:* побудовано автором

Загалом, Slashdot достатньо старий сервіс, який більше спеціалізується на розповсюдженні новин, але з специфічною модерацією.

Останнім в порівнянні є Code Project [28]. Це онлайн-спільнота розробників програмного забезпечення, яка надає ресурси, статті, уроки та форуми для програмістів та інженерів програмного забезпечення. Головна сторінка сервісу зображена на рисунку 1.5.

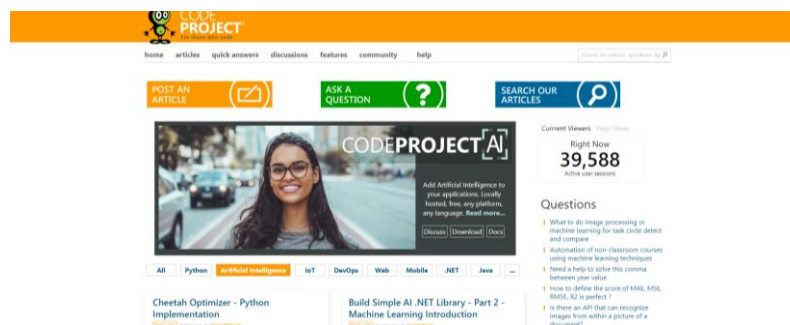


Рисунок 1.5 – Головна сторінка «Code Project»

*Джерело:* знімок з екрану [28]

Дизайн має мінімалістичну стилістику, а перехід на статтю одразу показує потрібну інформацію. Перегляд інформації не потребує реєстрації, а от для перегляду коментарів та проведення дискусії доведеться авторизуватися. Відповідь на розв'язання проблеми зображено на рисунку 1.6.

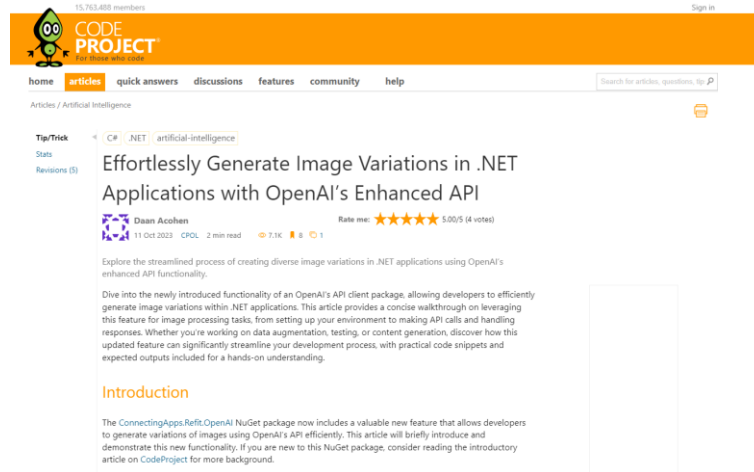


Рисунок 1.6 – Сторінка відповіді на питання на сайті «Code Project»

*Джерело:* знімок з екрану [28]

При перегляді тем одразу помічаєш вузьконаправлені теми. При детальному розгляді можна побачити застарілий код або поради для вирішення тих чи інших проблем. Відповідь на розв'язання проблеми зображено на рисунку 1.7.

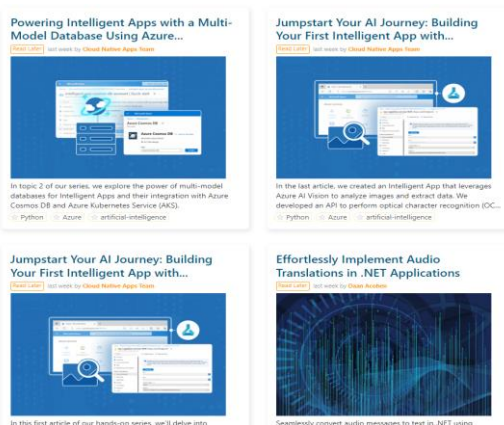


Рисунок 1.7 – Сторінка відповіді на питання на «Code Project»

*Джерело:* знімок з екрану [28]

Специфіка оформлення статті відштовхує нових користувачів та може здатися незвичною для користувачів інших сервісів. Також негативно впливає на сприйняття сервісу – відсутність темної теми. Відповідь на розв’язання проблеми зображено на рисунку 1.8.

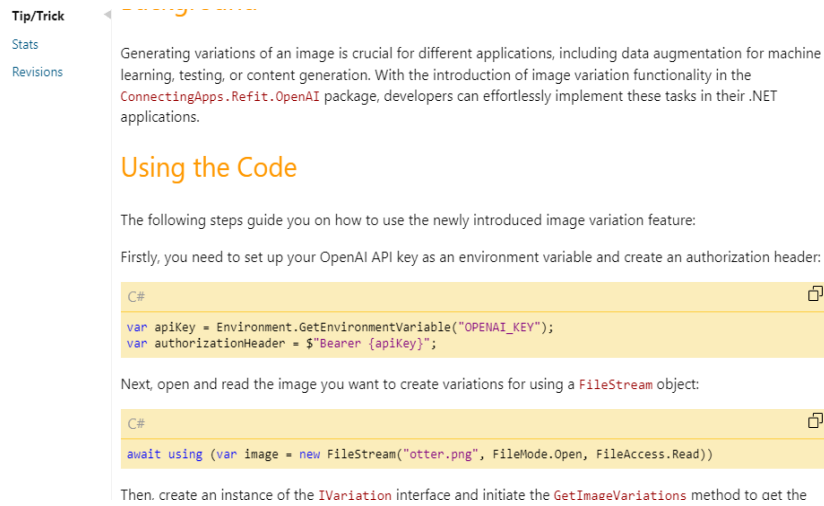


Рисунок 1.8 – Сторінка відповіді на питання на «Code Project»

Джерело: знімок з екрану [28]

В таблиці 1.3 представлені співставлені переваги та недоліки веб-додатку Code Project.

Таблиця 1.3 – Порівняльна таблиця переваг та недоліків «Code Project»

Переваги	Недоліки
Code Project насичений великою кількістю матеріалів, включаючи статті, уроки, кодові приклади та форуми, що може бути дуже корисним для програмістів з пошуком відповіді на свої питання;	Застарілість деяких матеріалів в Code Project не відображає останні технологічні тенденції та не надає оперативні дані про останні новинки;

Продовження таблиці 1.3

Переваги	Недоліки
Активна спільнота Code Project з радістю готова надавати допомогу та обговорювати технічні питання;	Code Project спеціалізується на розробці для Windows, і хоча тематика розширилася, переважна частина тем пов'язана з розробкою на Windows;
Різноманітність тем охоплює широкий спектр тем, включаючи розробку програмного забезпечення, web-розробку, бази даних, мобільний розробку та інші галузі;	Деякі матеріали містять помилки або застарілий код, що може призвести до проблем у розробці проекту;
Безкоштовний доступ дозволяє програмістам здобувати користь без необхідності оплати.	Специфічний стиль публікацій Code Project важко засвоюється новачком, бо потребує базовий рівень технічного досвіду.

*Джерело:* побудовано автором

Загалом Code Project хоч і має велику інформаційну базу та кодинг, але багато застарілого матеріалу, або того що потребує перевірки. Хоч відкритість системи сильно приваблює нових юзерів.

Таблиця 1.4 – Порівняльний аналіз аналогів

Критерії	«Stack Overflow»	«Slashdot»	«Code Project»	Web-орієнтована система консультування IT-фахівців з професійних питань
Приємний дизайн	+	–	–	+

Продовження таблиці 1.4

Критерії	«Stack Overflow»	«Slashdot»	«Code Project»	Web-орієнтована система консультування ІТ-фахівців з професійних питань
Охоплення різних напрямків	+	+	+	+
Достовірність інформації	+	–	–	+
Комфортна навігація	–	+	+	+
Інформативність відповідей	+	+	–	+
Швидкість отримання відповіді	+	–	–	+
Використання штучного інтелекту	–	–	–	+

*Джерело: побудовано автором*

Проаналізувавши конкурентні аналоги, можна зробити наступні висновки. Web-орієнтовані системи та сервіси мають свої унікальні функції, але також мають свої недоліки. Це обмежує їх спеціалізацію, призначення лише для певних галузей. Після оцінки точності та швидкості надання відповідей було підтверджено, що використання штучного інтелекту є важливим.

Отже, web-орієнтовані системи повинні бути універсальними в галузевому плані для професійних питань в ІТ-сфері, володіти високою швидкістю та точністю відповідей та забезпечувати комфортну роботу з доступом до всіх необхідних функцій.

## 2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

### 2.1. Мета та задачі дослідження

Мета кваліфікаційної роботи є розробити якісний додаток для консультування ІТ-фахівців у сфері професійних питань, де штучний інтелект використовується як складова частина інформаційного середовища для покращення взаємодії між фахівцем і користувачем у різних аспектах.

Для успішної реалізації web-орієнтованої системи потрібно зосередити увагу на всіх важливих аспектах, щоб основні функції були завжди легкодоступними та інтуїтивно зрозумілими для користувачів. Також буде розроблена та використана база для збереження та оброблення даних користувачів.

Для роботи із штучним інтелектом буде використана API Chat GPT. На момент розробки це є легка у використанні API із перевіреною часом системою відповіді на запитання.

Представлено перелік функціоналу, що потрібно реалізувати при розробці web-орієнтованої системи для звичайного користувача:

1. Реєстрація у web-орієнтованій системі;
2. Авторизація у web-орієнтованій системі;
3. Створення запитання;
4. Надання відповіді на запитання;
5. Зміна статусу на питання;
6. Отримання відповіді на запитання від ChatGPT;
7. Оцінка відповіді від іншого користувача;
8. Пошук та фільтрація запитань та відповідей від зареєстрованих користувачів;
9. Використання форуму зареєстрованими користувачами;
10. Створення теми для обговорення на форумі;
11. Написання коментарю в темі обговорення форуму;
12. Редагування особистих даних.



Представлено перелік функціоналу, що потрібно реалізувати при розробці web-орієнтованої системи для адміністратора:

1. Авторизація у web-орієнтованій системі;
2. Редагування інформації на сайті;
3. Публікація новин та оновлень web-орієнтованої системи;
4. Блокування користувачів;
5. Блокування обговорень;
6. Блокування статей.

Функціональні вимоги:

1. Безпека у використанні системи;
2. Безпечність при наданні особистої інформації користувачами;
3. Коректна роботи із API ChatGPT;
4. Збереження інформації, з якою працює користувач;
5. Шифрування особистих даних на стороні бази даних.

Не функціональні вимоги:

1. Надійність роботи на сервісу;
2. Легкість у використанні web-орієнтованої системи;
3. Портативність web-орієнтованої системи;
4. Зручність використання системою;
5. Сумісність із іншими середовищами.

Отже, було визначено перелік функціоналу, що потрібно реалізувати при розробці web-орієнтованої системи. Також було сформовано функціональні та нефункціональні вимоги.

## **2.2. Розробка візуальної частини web-орієнтованої системи консультування**

Розробники завжди шукають найкращі інструменти та технології для використання в проєкт. Для розробки візуальної частини web-орієнтованої системи консультування було обрано зосередитися на двох інструментах: Tailwind CSS і Vue.js.

Фреймворк Vue.js забезпечує декларативну модель програмування на основі компонентів, яка допомагає створювати як прості, так і складні інтерфейси користувача [29]. Коли справа доходить до взаємодії різних компонентів, які зараз відображаються серверами, цей фреймворк вважається надійним варіантом.

Крім цього існує можливість зв'язування даних у Vue.js, що дозволяє вказувати елементи, значення та оновлення. Ці дії дозволяють отримати точні результати та отримати максимум від процедури прив'язки даних.

Також важливим аспектом є те, що Vue.js має менший розмір фінальної програми. Фреймворк виділяється серед інших фреймворків, завдяки своєму компактному розміру. Зазвичай розмір програми Vue.js становить від 18 до 21 КБ, не враховуючі дані у вигляді файлів чи зображень, вже доданими користувачами.

Найкраща особливість полягає в тому, що у даному фреймворку присутньо більше свободи створювати нові програми або змінювати вже написані. Однією з найбільш рекламованих переваг Vue.js є його гнучкість.

Як відомо, Tailwind CSS перш за все утиліта CSS-фреймворку, яка надає широкий спектр попередньо розроблених класів [30]. Однією з головних переваг Tailwind CSS надання мінімального набору стилів, які можна легко налаштувати та розширити, не будучи обмеженим фреймворком. Це робить його ідеальним вибором для швидкої та якісної розробки.

Саме поєднання Tailwind CSS і Vue.js, та їх простота, гнучкість, продуктивність, роблять їх ідеальним вибором для створення інтерфейсів користувача [31], які будуть швидкими, надійними та простими у використанні.

Також при розробці web-орієнтованої системи консультування було використано додаткову бібліотеку Flowbite. Дана бібліотека являти собою набір компонентів на основі Tailwind CSS, який можна просто скопіювати та вставити з документації.

Він також містить файл JavaScript, який дає змогу використовувати інтерактивні компоненти, такі як моди, розкривні меню та засоби вибору дати, які ви можете додати до свого проєкт за допомогою NPM.

Дана бібліотека розширює перелік можливих компонентів для використання, як, наприклад:

- Оповіщення;
- Значки;
- Кнопки;
- Група кнопок;
- Випадаючі списки;
- Форми;
- Типографіка;
- Модальні вікна;
- Пагінація.

Іншим інструментом в розробці візуальної частини web-орієнтованої системи консультування є Syntactically Awesome Stylesheets (SASS). Це препроцесор CSS, який дозволяє використовувати змінні, математичні операції, цикли, функції та інші функціонал, який також розширює можливості CSS та робить його набагато потужнішим [32-33].

Як додатковий плагін був використаний SKEditor5 [34]. Даний функціонал пропонує будь-який тип рішення для редагування. Редактор має добре розроблений інтерфейс користувача та ідеальний UX, тому користувачі можуть легко керувати медіафайлами та таблицями, а також використовувати розширені функції, як-от автоматичне форматування, згадування та інше.

Інший інструмент, що був використаний це Vite. Іншими словами це інструмент збірки, представлений розробниками Vue.js. Даний функціонал використовує сучасний модульний підхід до створення додатків та розроблено для використання переваг сучасних функцій браузера.

Головні переваги та функціонал, що надає Vite:

- Швидший час створення;
- Повна інтеграція;
- Покращене налагодження.

Отже, обраний інструментарій для розробки візуальної частини web-орієнтованої системи консультування, а саме Tailwind CSS і Vue.js, SASS, Flowbite, SKEditor5, Vite, є чудовими варіантами, що чудово поєднуються та охоплюють увесь необхідний функціонал, для реалізації додатка.

### **2.3. Розробка функціональної частини web-орієнтованої системи консультування**

Як відомо розробникам, Laravel є відомим та популярним фреймворк із відкритим кодом. Кожна нова версія Laravel вдосконалюється новими веб-рішеннями.

Laravel дозволяє створювати багатофункціональні сайти та програми завдяки створеним пакетам і функціям, наприклад як системи безпеки та сайтовий захист від копіювання інформації [35]. Також присутній функціонал, де реалізована логіка автентифікації користувачів, спрощена інтеграція з програмами електронної пошти, система черги повідомлень і незавершені завдання.

Адже безпека це питання номер один для більшості компаній. За допомогою Laravel розширені функції безпеки можна легко налаштувати, щоб покращити безпеку та захист сайтів від хакерів. Щоб отримати трохи технічної інформації, Laravel використовує алгоритм хешування bcrypt, що означає, що він ніколи не зберігає паролі в базі даних.

Головними перевагами використання даного фреймворку є:

- Швидке впровадження нового функціоналу на веб-ресурсі, не витрачаючи зайвих ресурсів;
- Впевненість у тому, що сайт або додаток можна легко оновити.
- Гнучкість, адже Laravel дозволяє масштабувати сайти та програми будь-якої складності та розміру;
- Можливість реалізувати будь-яку ідею за допомогою функціоналу фреймворку;
- Контролювання доступу до захищених ресурсів;

- Легкість у обробці запитів на доступ;
- Можливість відхиляти неавторизовані запити.

Laravel підтримує структуру MVC (Model-View-Controller), яка автоматично забезпечує поділ між логікою та синтаксисом виразу. Це розділення полегшує розробникам інтерфейсу вдосконалення інтерфейсу користувача, не порушуючи розробку чи основні функції. Крім того, розділення гарантує, що розробники можуть легко додавати нові функції та виправляти помилки, витрачаючи мінімум часу.

Інтеграція Laravel значно полегшує життя розробникам [36]. Цей інструмент дозволяє створювати базові файли MVC і генерувати власні команди під час створення базового коду, архітектури бази даних і міграції.

Отже, обраний фреймворк для розробки функціональної частини web-орієнтованої системи консультування, а саме Laravel, є чудовим варіантом, що має необхідний функціонал, для реалізації додатка.

## 3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

### 3.1 Структура web-орієнтованої системи консультування

Важливим етапом розробки web-орієнтованої системи консультування є проектування структури. Базуючись на переліку функціонала системи, можна сформулювати список необхідних сторінок. Розглянемо детальніше в таблиці 3.1.

Таблиця 3.1 – Перелік сторінок системи

№	Назва сторінки	Доступ користувача до сторінки	
		Зареєстрований	Незареєстрований
1	Авторизація	+	+
2	Реєстрація	+	+
3	Головна	+	+
4	Запитання	+	–
5	Детальна інформація з обраним запитанням	+	–
6	Користувачі	+	–
7	Детальна інформація про обраного користувача	+	–
8	Статті	+	–
9	Детальна інформація по обраній статі	+	–
10	Форум	+	–
11	Детальна інформація про обрану тему обговорення на форумі	+	–
12	Додання власного питання	+	–
13	Створення власної теми обговорення	+	–

Продовження таблиці 3.1

№	Назва сторінки	Доступ користувача до сторінки	
		Зареєстрований	Незареєстрований
14	Редагування особистої інформації	+	–
15	Запитання користувача	+	–
16	Статті користувача	+	–
17	Теми користувача	+	–

*Джерело: побудовано автором*

Окрім цього було додатково спроектовано перелік сторінок для адміністративної панелі. Адміністратор матиме додаткові сторінки, а саме:

- Користувачі;
- Статі;
- Теми форуму;
- Питання;
- Теги;
- Заборонені слова;

Отже, для реалізації та використання усього функціоналу web-орієнтованої системи консультування було розроблено перелік сторінок, що необхідно розробити. Структура розрахована на якісну взаємодію із користувачами системи.

### 3.2 Структурно-функціональне моделювання

Для web-орієнтованої системи поставлена конкретна задача – створити інформаційну систему для консультування IT-фахівців з професійних питань. Для побудови буде використано вхідні та вихідні дані, механізми та управління [37].

Вхідними даними є: інформація про користувача, запитання до системи. Вихідними даними є: відповідь питання, підвищення рейтингу фахівця. Механізмами є: розробник, програмне забезпечення, web-сервіс, апаратне забезпечення, API Chat GPT. Контекстна діаграма показана на рисунку 3.1.

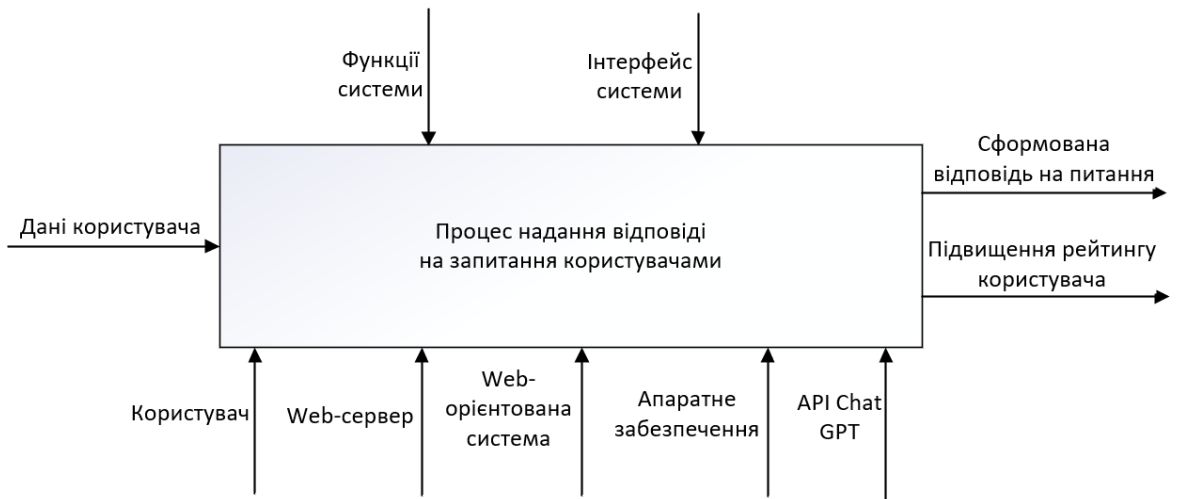


Рисунок 3.1 – Контекстна діаграма web-орієнтованої системи по процесу надання відповіді на запитання з використанням API Chat GPT

*Джерело: побудовано автором*

Модель web-орієнтовані системи для консультування ІТ-фахівців з професійних питань показано у вигляді діаграми декомпозиції на рисунку 3.2.

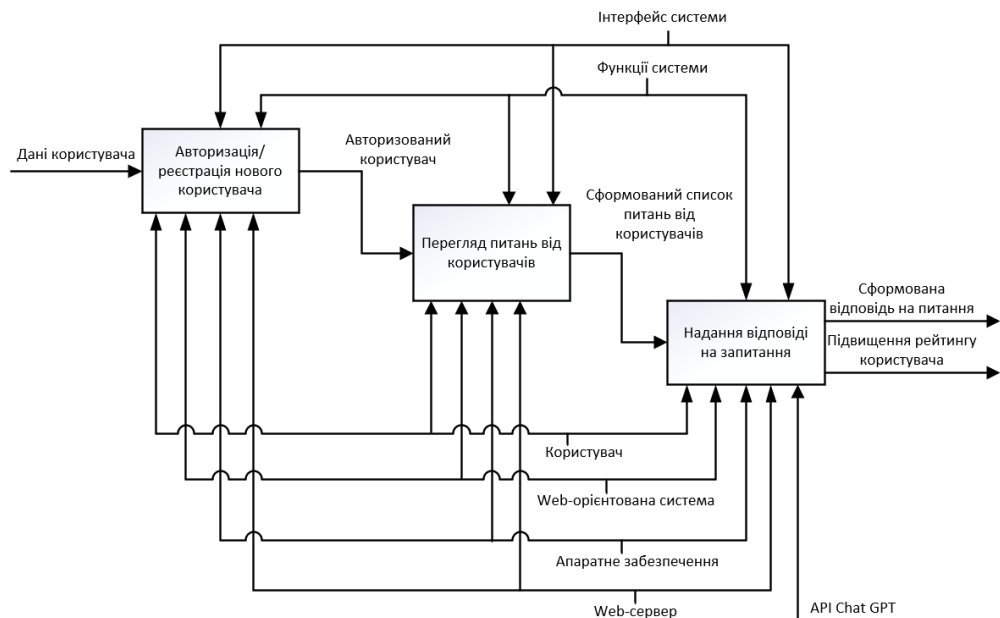


Рисунок 3.2 – Декомпозиція контекстної діаграми по процесу надання відповіді на запитання з використанням API Chat GPT

*Джерело: побудовано автором*



Також була розроблена контекстна діаграма по процесу створення запитання користувачем (рис.3.3).

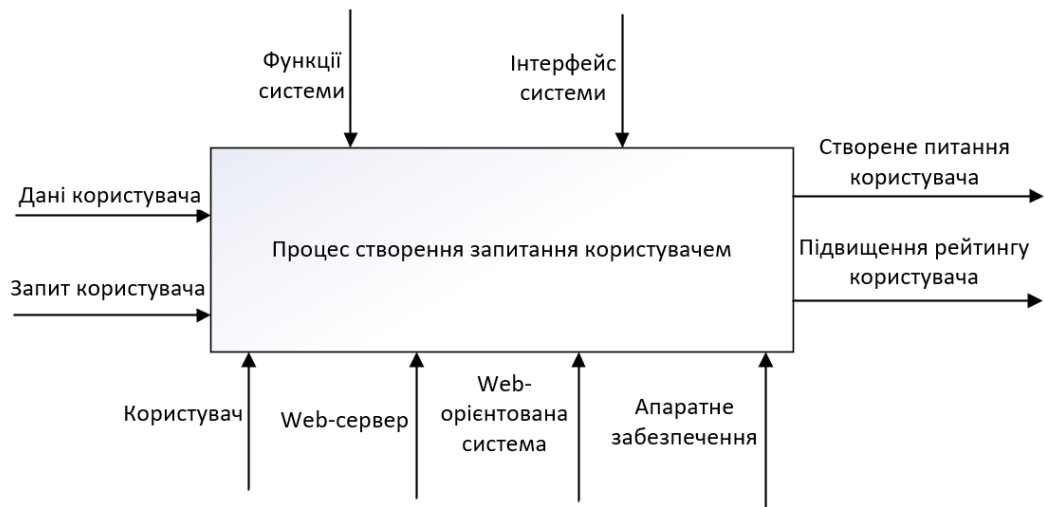


Рисунок 3.3 – Контекстна діаграма web-орієнтованої системи по процесу створення запитання користувачем

*Джерело: побудовано автором*

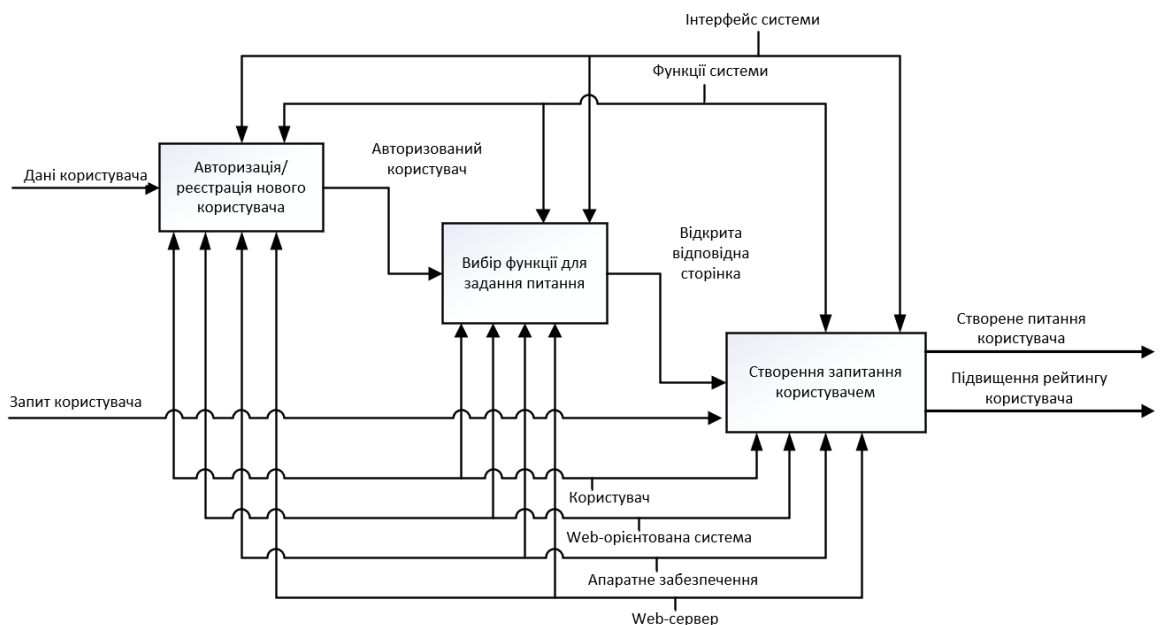


Рисунок 3.4 – Декомпозиція контекстної діаграми web-орієнтованої системи по процесу створення запитання користувачем

*Джерело: побудовано автором*

### 3.3 Моделювання діаграми варіантів web-орієнтованої системи консультування

Маючи уявлення про функціональні можливості web-орієнтованих систем для консультування IT-фахівців з професійних питань, була сформована схема варіантів використання для користувача (К) та адміністратора (А), що представлено на рисунку 3.5 та 3.6.

Детальний опис акторів та варіантів використання представлено в таблиці 3.2 та таблиці 3.3.

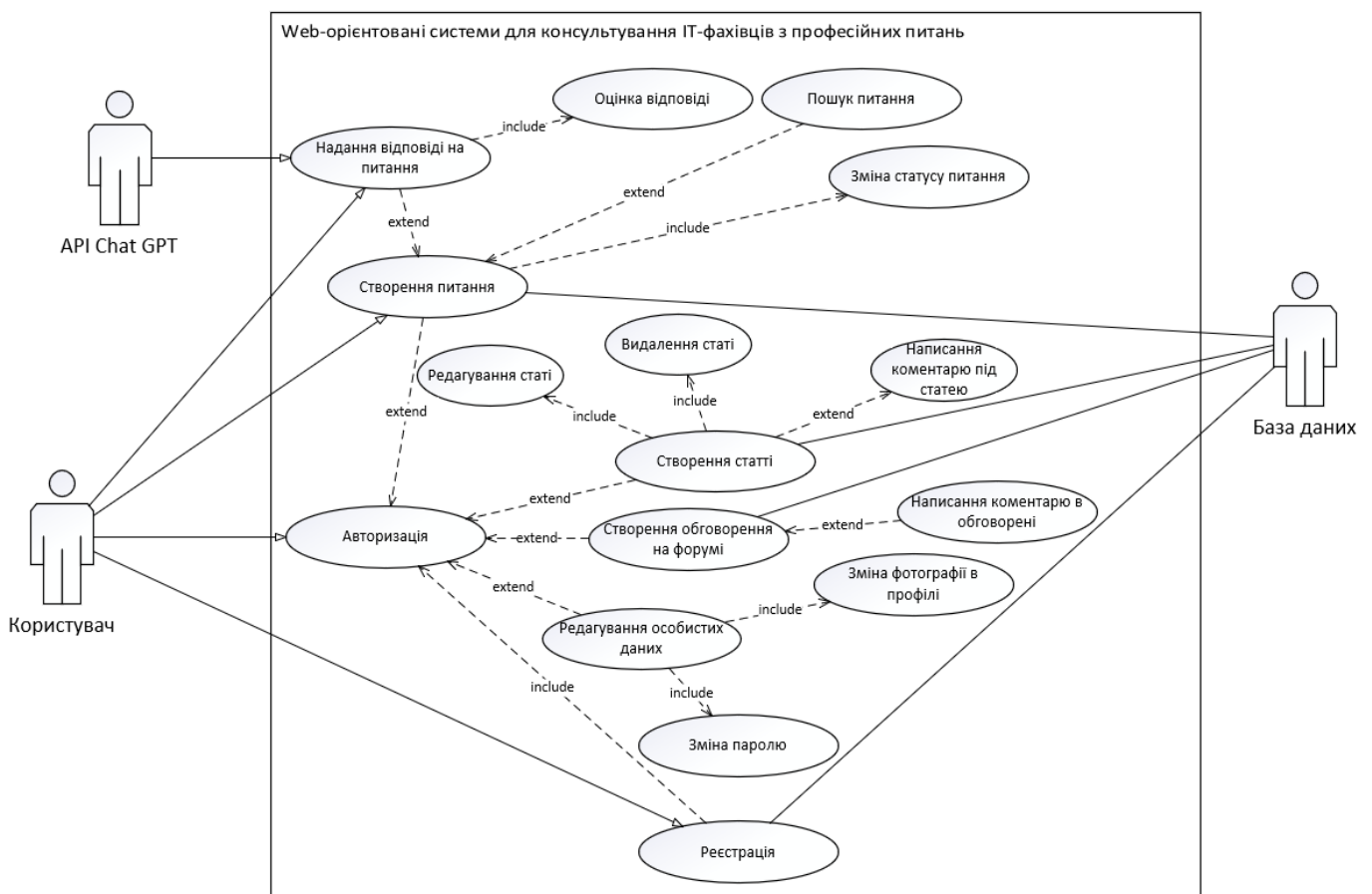


Рисунок 3.5 – Діаграма варіантів використання для користувача

Джерело: побудовано автором



Таблиця 3.3 – Опис варіантів використання

№	Варіанти використання	Детальний опис	Актор
1	Авторизація	Функція для авторизації	К/А
2	Реєстрація	Функція для реєстрації	К/А
3	Створення питання	Користувач може поставити питання та отримати відповідь на нього	К
4	Надання відповіді на питання	Користувачі мають можливість відповідати на запитання інших користувачів	К
5	Зміна статусу питання	Користувач може змінювати статус свого питання	К
6	Відповідь від API Chat GPT	Користувачі мають можливість отримати відповідь від Chat GPT	К
7	Оцінка відповіді	Користувач може оцінювати відповіді на питання	К
8	Пошук запитань	Користувачі можуть шукати запитання з використанням фільтрації	К
9	Створення обговорення на форумі	Користувач може створювати обговорення на форумі	К
10	Написання коментаря в обговоренні	Користувач може писати коментарі в обговореннях	К
11	Редагування особистих даних	Зареєстровані користувачі можуть редагувати особисті дані	К
12	Зміна фотографії в профілі	Зареєстровані користувачі можуть змінювати фотографії в профілі	К
13	Зміна пароля	Зареєстровані користувачі можуть змінювати пароль	К

Продовження таблиці 3.3

№	Варіанти використання	Детальний опис	Актор
14	Написання статі	Зареєстровані користувачі можуть створювати власні статі	К
15	Написання коментаря під статтями	Зареєстровані користувачі можуть писати коментарі в статтях	К
16	Редагування статі	Користувачі можуть редагувати власні статті	К
17	Видалення статі	Користувачі можуть видаляти власні статті	К
18	Редагування інформації на сайті	Адміністратор може редагувати основну інформацію на сайті	А
19	Модерування форумів	Адміністратор контролює інформацію, що перебуває на форумі	А
20	Блокування коментаря	Адміністратор може видаляти коментарі в статтях	А
21	Блокування теми форуму	Адміністратор може блокувати теми форуму	А
22	Блокування користувача	Адміністратор може блокувати користувачів, що порушують правила використання додатка	А
23	Публікація новин	Адміністратор може публікувати новини та оновлення	А

### 3.4. Проектування бази даних

Основні користувачі системи – фахівець та користувач. Вони мають змогу отримувати різні відомості про консультації IT-фахівців з професійних питань.

Показана візуалізація на ER-діаграмі (рис.3.4). Для проектування бази даних було проаналізовано статті, де були представлені шляхи вирішення та кращі варіанти при розробці та моделюванні [38]. У цій статті обговорюється як функціональність, так і відносні сильні та слабкі сторони кожного із підходів проектування бази даних.

Було виконано перехід до моделювання бази даних, що виконується після структурного аналізу. Перелік полів таблиць бази даних показано в таблиці 3.4.

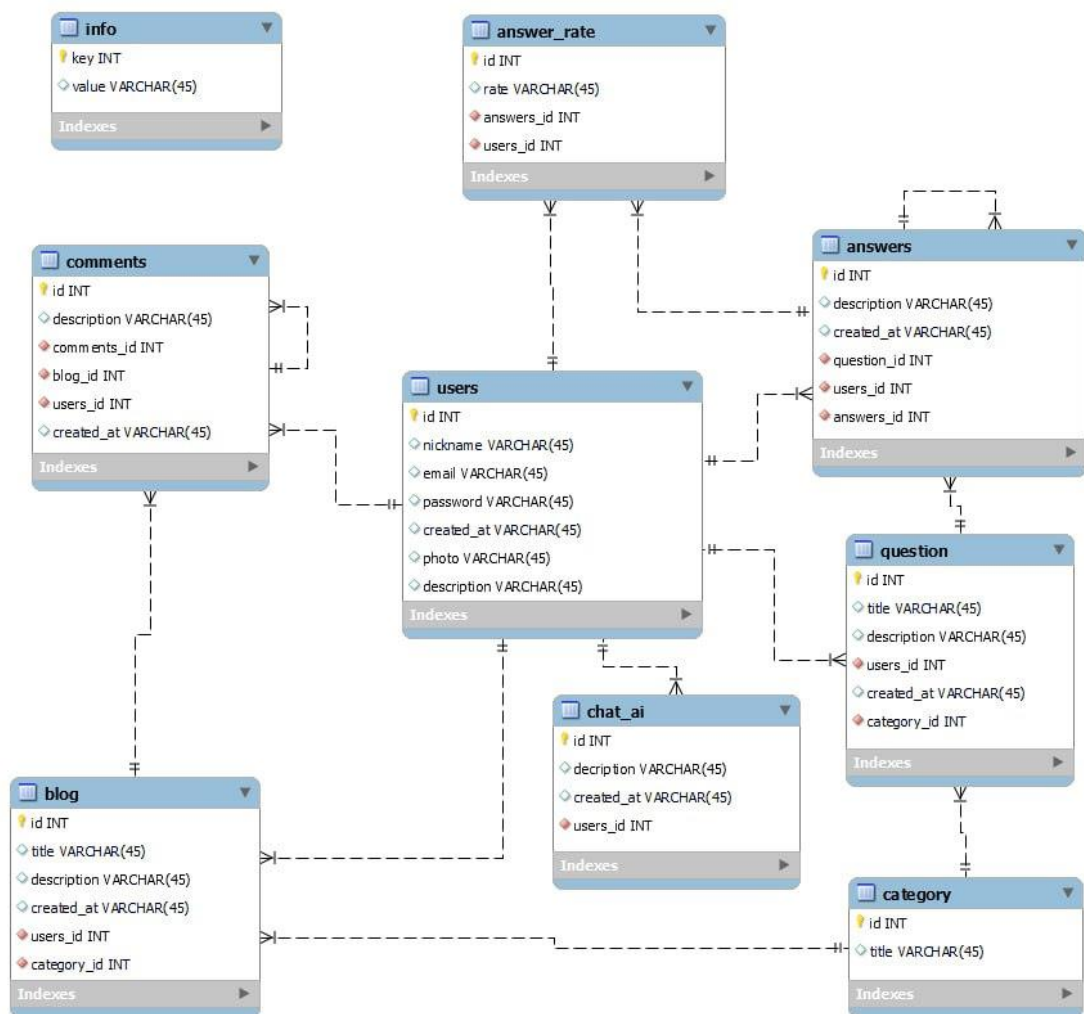


Рисунок 3.5 – ER-діаграма

Джерело: побудовано автором

Таблиця 3.4 – Перелік полів таблиць ER-діаграми web-орієнтованої системи консультування IT-фахівців з професійних питань

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
1	sp_users	ip_id	Ідентифікатор аккаунта	INTEGER	PK	Не порожній
		sp_nickname	Ім'я користувача	VARCHAR(45)		Не порожній
		sp_email	Електронна пошта	VARCHAR(45)		Не порожній
		sp_password	Пароль користувача	VARCHAR(45)		Не порожній
		sp_created_at	Дата створення	VARCHAR(45)		Не порожній
		sp_photo	Фото	VARCHAR(45)		Не порожній
		sp_description	Опис аккаунта	VARCHAR(45)		Не порожній
2	sp_answer_rate	sp_id	Ідентифікатор відгуку	INTEGER	PK	Не порожній
		sp_rate	Оцінка	VARCHAR(45)		Не порожній
		sp_answers	Коментар відповіді	INTEGER	FK	Не порожній
		sp_users_id	Ідентифікатор користувача	INTEGER	FK	Не порожній
3	sp_answers	sp_id	Ідентифікатор відповіді	INTEGER	PK	Не порожній
		sp_description	Опис відповіді	VARCHAR(45)		Не порожній
		sp_created_at	Дата створення	VARCHAR(45)		Не порожній
		sp_question_id	Ідентифікатор запитання	INTEGER	FK	Не порожній
		sp_users_id	Ідентифікатор користувача	INTEGER	FK	Не порожній
		sp_answers_id	Ідентифікатор коментаря зустрічі	INTEGER	FK	Не порожній

Продовження таблиці 3.4

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
4	sp_question	sp_id	Ідентифікатор питання	INTEGER	PK	Не порожній
		sp_title	Назва категорії	VARCHAR(45)		Не порожній
		sp_description	Опис відповіді	VARCHAR(45)		Не порожній
		sp_users_id	Ідентифікатор аккаунта	INTEGER	FK	Не порожній
		sp_created_at	Дата створення	VARCHAR(45)		Не порожній
		sp_category_id	Ідентифікатор категорії	INTEGER	FK	Не порожній
5	sp_category	sp_id	Ідентифікатор категорії	INTEGER	PK	Не порожній
		sp_title	Назва категорії	VARCHAR(45)		Не порожній
6	sp_shat_ai	sp_id	Ідентифікатор запиту	INTEGER	PK	Не порожній
		sp_description	Опис запиту	VARCHAR(45)		Не порожній
		sp_created_at	Дата створення	VARCHAR(45)		Не порожній
		sp_users_id	Ідентифікатор користувача	INTEGER	FK	Не порожній
7	sp_blog	sp_id	Ідентифікатор блогу	INTEGER	PK	Не порожній
		sp_title	Назва блогу	INTEGER		Не порожній
		sp_description	Опис блогу	VARCHAR(45)		Не порожній
		sp_created_at	Дата створення	VARCHAR(45)		Не порожній
		sp_users_id	Ідентифікатор аккаунта	INTEGER	FK	Не порожній
		sp_category_id	Ідентифікатор категорії	INTEGER	FK	Не порожній



## Продовження таблиці 3.4

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
8	sp_comments	sp_id	Ідентифікатор коментаря	INTEGER	PK	Не порожній
		sp_description	Опис коментаря	VARCHAR(45)		Не порожній
		sp_comments_id	Ідентифікатор коментарів	INTEGER	FK	Не порожній
		sp_blog_id	Ідентифікатор блогу	INTEGER	FK	Не порожній
		sp_users_id	Ідентифікатор користувача	INTEGER	FK	Не порожній
		sp_created_at	Дата створення	VARCHAR(45)		Не порожній
14	sp_info	sp_key	Ідентифікатор запису	INTEGER	PK	Не порожній
		sp_value	Службова інформація	VARCHAR(45)		Не порожній

*Джерело:* побудовано автором

## 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ КОНСУЛЬТУВАННЯ

### 4.1 Програмна реалізація

Програмна реалізація web-орієнтованої системи з консультування виконана за допомогою таких технологій:

- Vue.js для реалізації візуальної частини додатка;
- Laravel, для розробки функціональної частини.

Дане поєднання є вже перевіреним часом варіантом, що використовується при розробці багатьох додатків [39]. Додаток був розроблений у Visual Studio Code. Структура проекту представлена на рисунку 4.1.

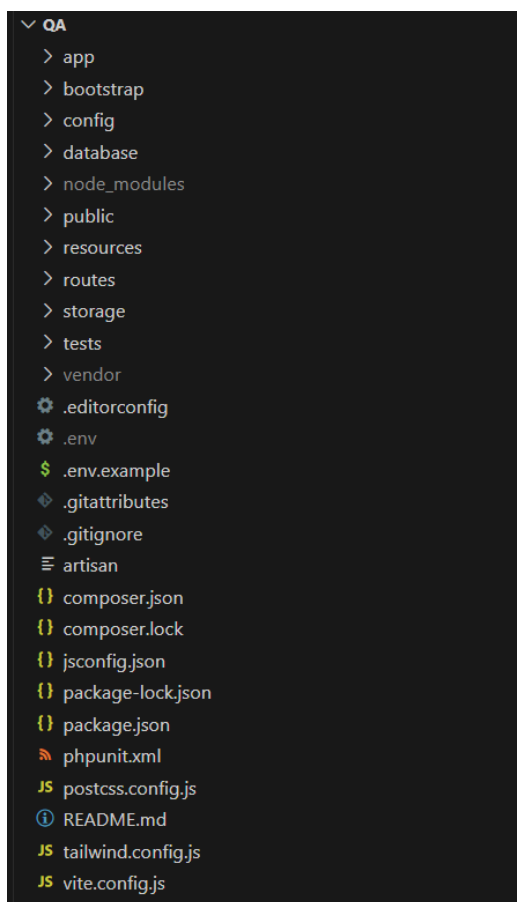


Рисунок 4.1 – Структура web-орієнтованої системи з консультування

*Джерело:* побудовано автором

Також було розроблено та використано базу даних, на основі phpMyAdmin та бази даних MySQL. Розроблена база даних представлена на рисунку 4.2.

На рисунку зображені всі таблиці за зв'язки між ними. Крім цього, розроблена база даних відповідає розробленій попередньо логічній моделі.

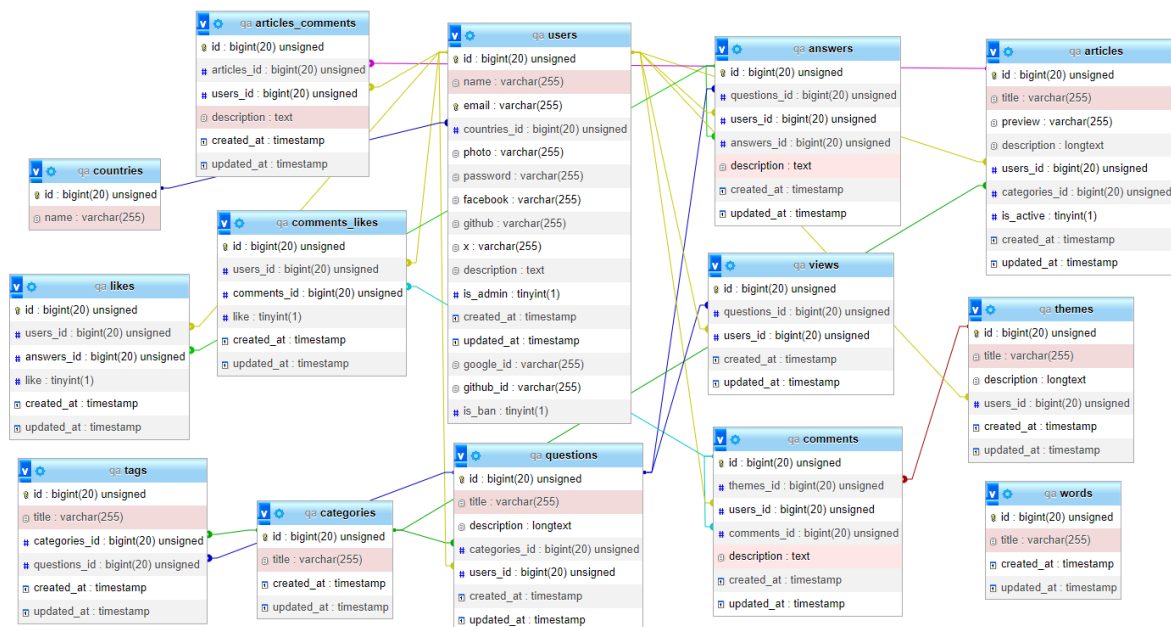


Рисунок 4.2 – Розроблена база даних в phpMyAdmin

*Джерело: побудовано автором*

Детальну інформацію про кожен із представлених таблиць можна переглянути у Додатку Б.

## 4.2 Використання web-орієнтованої системи користувачем

Для початку використання web-орієнтованої системи консультування потрібно перейти за посиланням на web-додаток [40]. Перейшовши за посиланням, користувач перенаправляється на головну сторінку.

Головна сторінка складається із декількох блоків, які несуть здебільшого інформативний контент. На рисунку 4.3 представлено першу частину сторінок. На

ній можна переглянути меню, загальну інформацію про кількість користувачів, кількість питань, відповідей та створених тем.

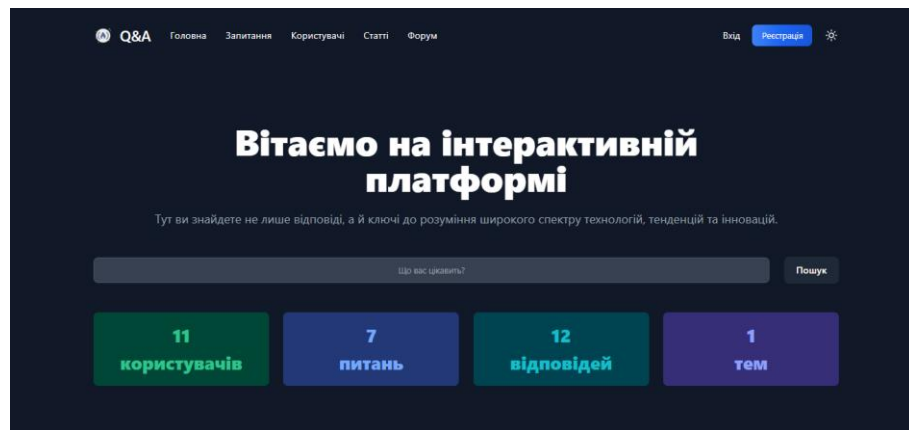


Рисунок 4.3 – Перша частина головної сторінки

*Джерело:* знімок з екрану [40]

Друга та третя частини головної сторінки містять в себе загальну теоретичну інформацію про використання штучного інтелекту на сьогоднішній час (рис.4.4).

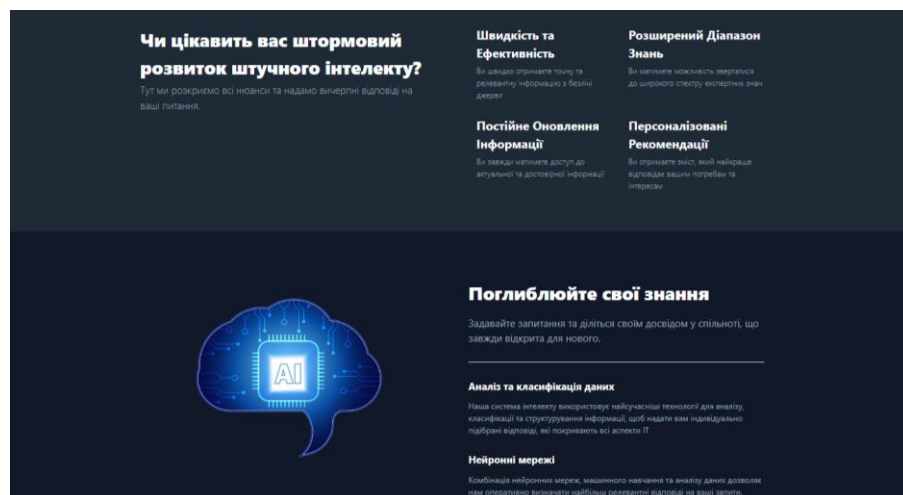


Рисунок 4.4 – Друга та третя частини головної сторінки

*Джерело:* знімок з екрану [40]

Наступна частина головної сторінки складає з себе перелік користувачів (рис.4.5), що мають найбільшу активність у web-орієнтованій системі консультування. Відбір до даного переліку відбувається за допомогою калькуляції

загальної кількості відповідей та сформованих статей. Для наповнення системи використано не реальні дані, а тестові. Додані фото для акаунтів було згенеровано у відповідних системах.

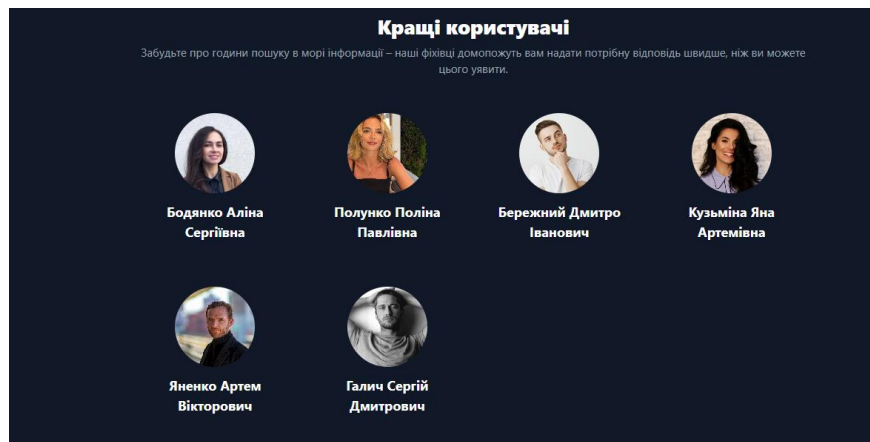


Рисунок 4.5 – Четверта частина головної сторінки

*Джерело:* знімок з екрану [40]

Остання п'ята частина головної сторінки складається зі списку питань та останніх наданих відповідей на них (рис.4.6). У цій частині відображаються відповіді, надані як за допомогою ChatGTP, так й звичайними користувачами.

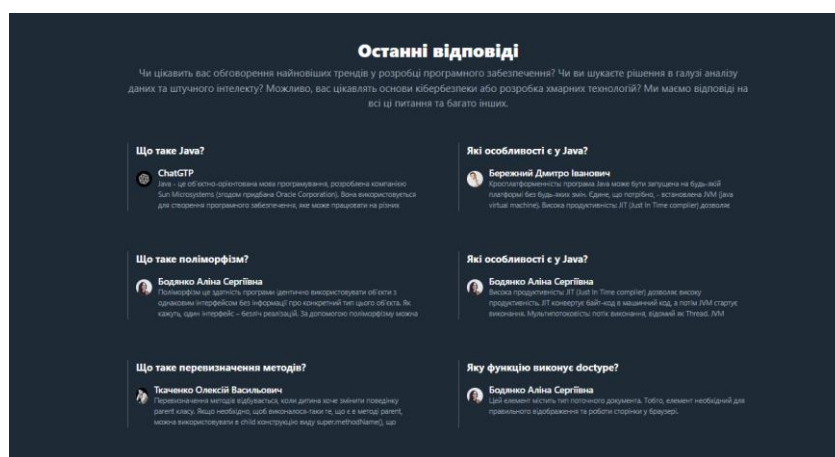


Рисунок 4.6 – П'ята частина головної сторінки

*Джерело:* знімок з екрану [40]

Переходячи на будь-яку іншу сторінку, неавторизований користувач перенаправляється на сторінку авторизації. Приклад представлений на рисунку 4.7.

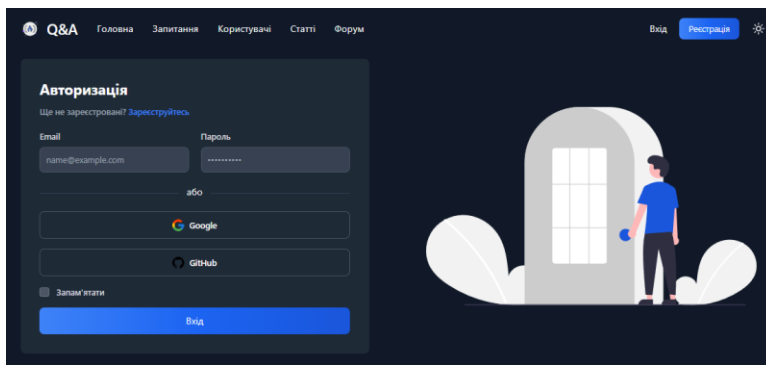


Рисунок 4.7 – Сторінка авторизації

*Джерело:* знімок з екрану [40]

Незареєстрований користувач може зареєструватися на відповідній сторінці (рис.4.8). Крім цього, для реєстрації та авторизації можна використовувати акаунт Google чи GitHub (рис.4.9-10).

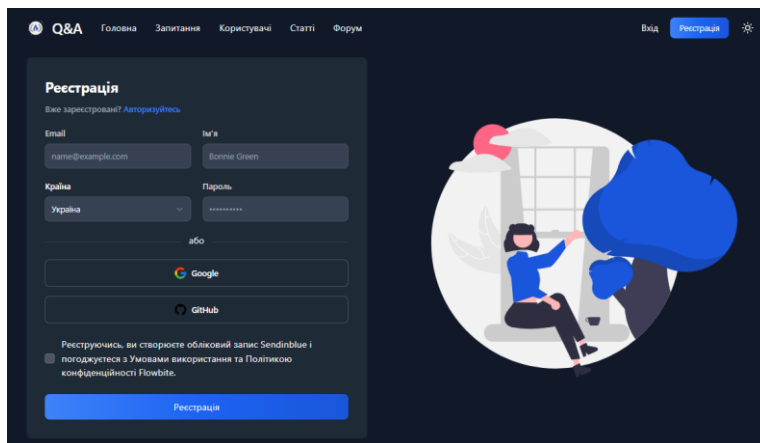


Рисунок 4.8 – Сторінка реєстрації

*Джерело:* знімок з екрану [40]

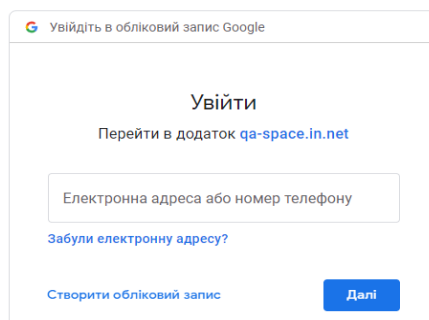


Рисунок 4.9 – Реєстрація за допомогою Google

*Джерело:* знімок з екрану [40]

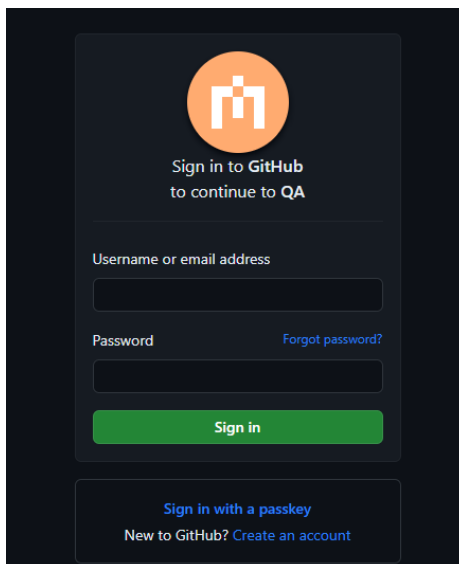


Рисунок 4.10 – Реєстрація за допомогою GitHub

*Джерело:* знімок з екрану [40]

Також було розроблено додатковий функціонал для зміни головної теми додатка. У правому верхньому куті можна обрати світлу чи темну тему web-орієнтованій системі консультування (рис.4.11). Темна тематика стоїть як початкова на сторінці.

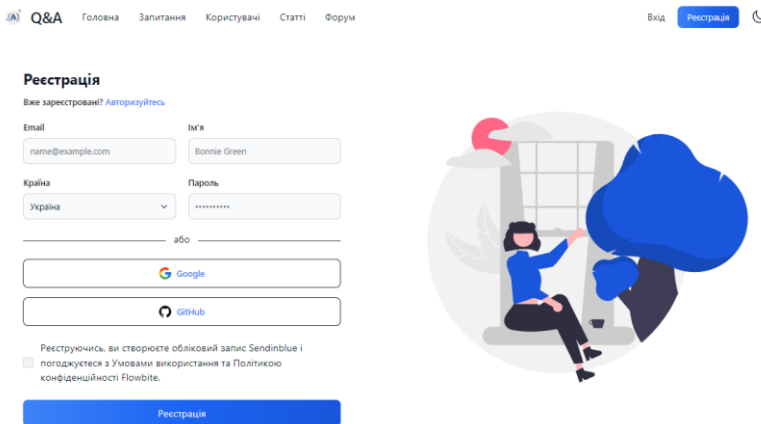


Рисунок 4.11 – Зміна теми web-орієнтованої системи

*Джерело:* знімок з екрану [40]

Після авторизації чи реєстрації у системи користувач має можливість переглянути сторінки із запитаннями, користачами та обговореннями на форумі.

Розглянемо детальніше сторінку із запитаннями (рис.4.12).

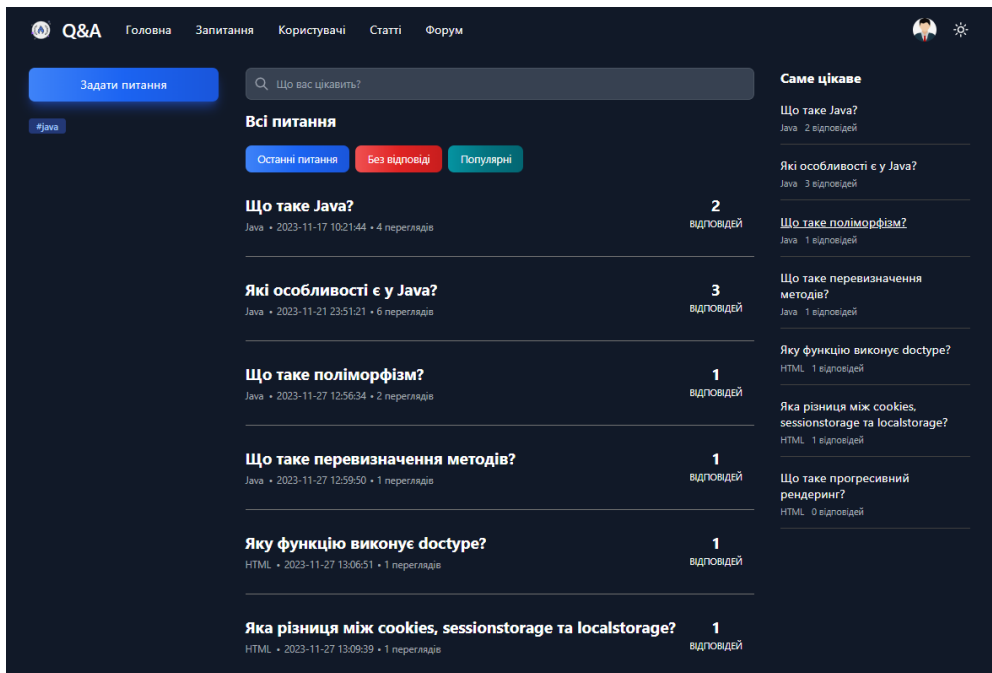


Рисунок 4.12 – Сторінка із запитаннями користувачів

*Джерело:* знімок з екрану [40]

На сторінці із запитаннями користувач має можливість переглянути всі питання створені у додатку. Також справа від запитань відображається актуальні та найпопулярніші запитання.

Зліва від усіх запитань відображається хмара з хештегів та фільтрація запитань по категоріях. Приклади виконання фільтрацій, а саме за популярністю, за кількістю відповідей та хештегами, представлені на рисунках 4.13 – 15.

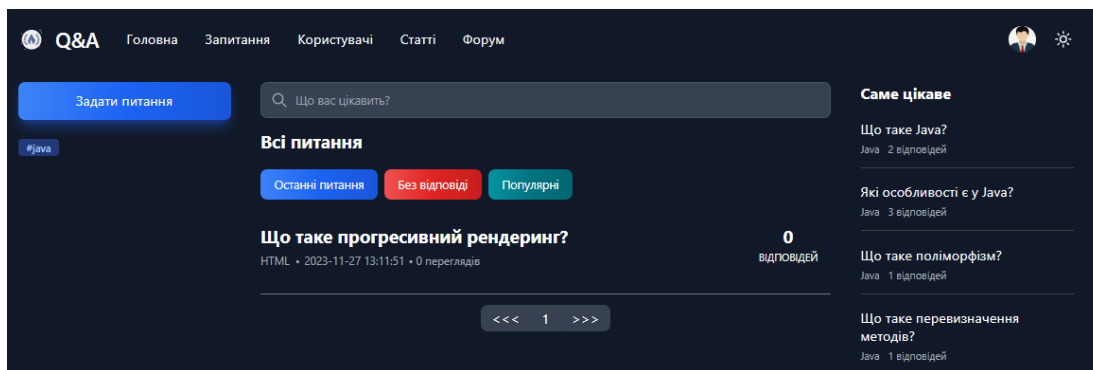


Рисунок 4.13 – Використання фільтрації «Без відповіді»

*Джерело:* знімок з екрану [40]



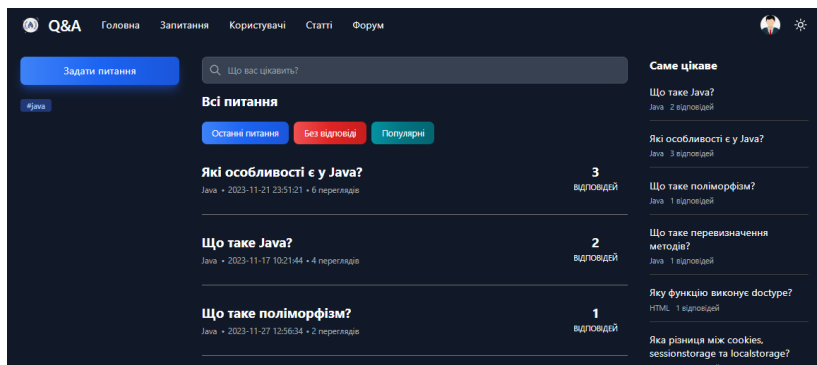


Рисунок 4.14 – Використання фільтрації «Популярні»

Джерело: знімок з екрану [40]

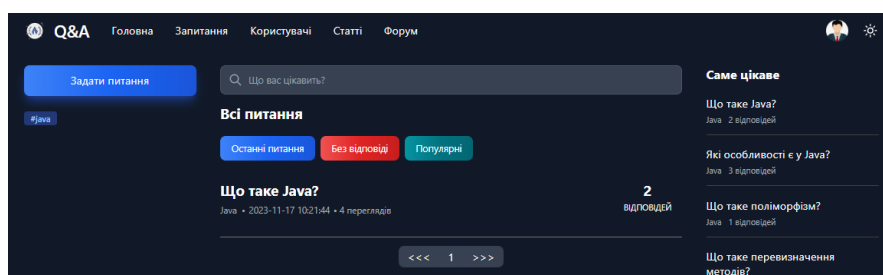


Рисунок 4.15 – Використання фільтрації за хештегом

Джерело: знімок з екрану [40]

Після використання фільтрації для пошуку питання користувач має можливість переглянути детальну інформацію по обраному питанню. Приклад перегляду інформації на рисунку 4.16.

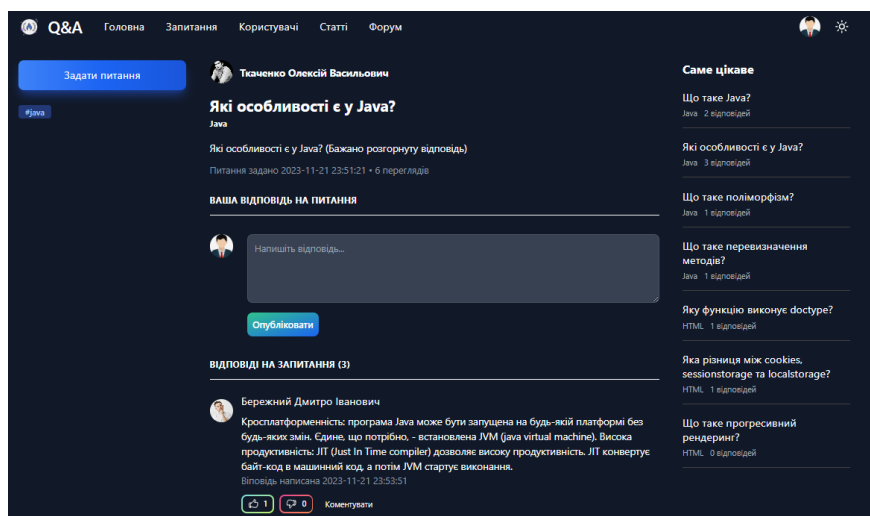


Рисунок 4.16 – Детальний перегляд обраного питання

Джерело: знімок з екрану [40]

При перегляді питання користувач може ознайомитися із питанням та його детальним описом. Крім цього, можна переглянути вже надані відповіді іншими користувачами. Присутня можливість опублікувати власну відповідь на запитання.

Також можна коментувати вже надані відповіді для уточнення інформації чи надання більш коректної відповіді (рис.4.17). Коментарі можна оцінити: позитивно чи негативно.

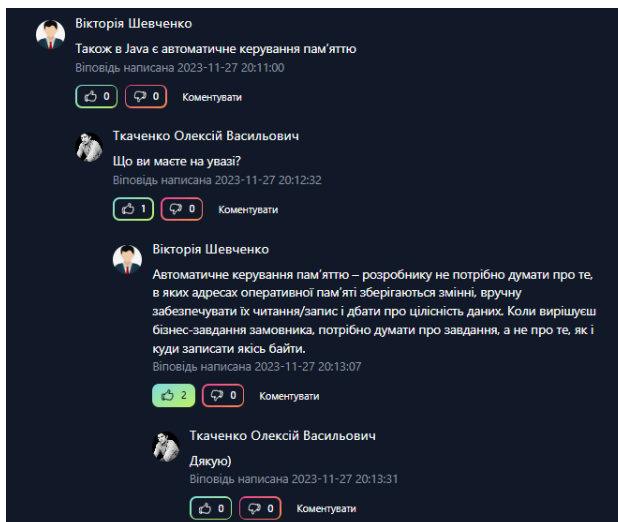


Рисунок 4.17 – Дерево із відповідей користувачів

*Джерело:* знімок з екрану [40]

Після усіх коментарів обраного питання користувач може переглянути схожі запитання та відповіді. Приклад на рисунку 4.18.

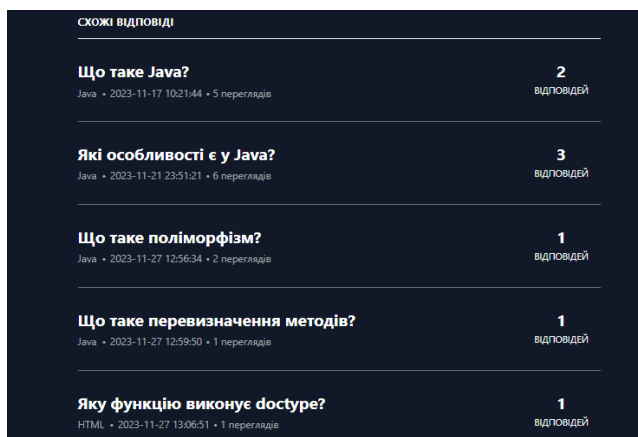


Рисунок 4.18 – Додаткова інформація про схожі питання

*Джерело:* знімок з екрану [40]

Будь-який зареєстрований користувач має можливість створити власне питання. Для цього необхідно натиснути кнопку «Задати запитання». Після натискання, користувач перенаправляється на сторінку для створення власного запитання (4.19).

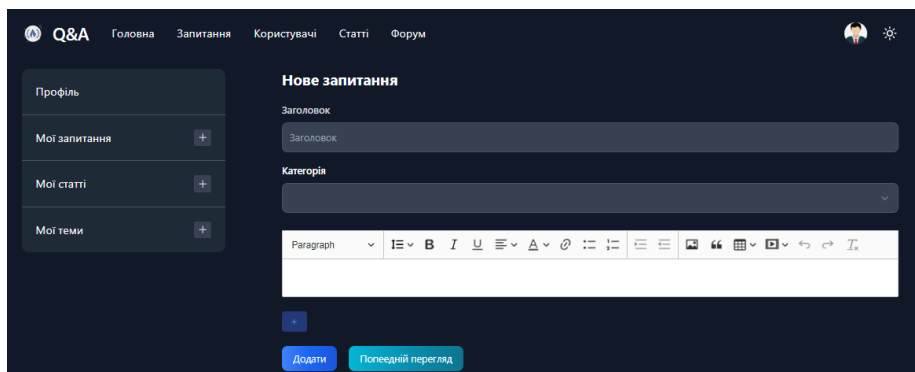
The image shows a dark-themed web interface for creating a new question. At the top, there is a navigation bar with 'Q&A' and links for 'Головна', 'Запитання', 'Користувачі', 'Статті', and 'Форум'. On the left, a sidebar contains 'Профіль', 'Мої запитання +', 'Мої статті +', and 'Мої теми +'. The main area is titled 'Нове запитання' and contains a 'Заголовок' input field, a 'Категорія' dropdown menu, and a rich text editor with a toolbar. At the bottom, there are 'Додати' and 'Попередній перегляд' buttons.

Рисунок 4.19 – Форма для створення власного запитання

*Джерело:* знімок з екрану [40]

На рисунку 4.20 представлено приклад заповнення даних для створення запитання.

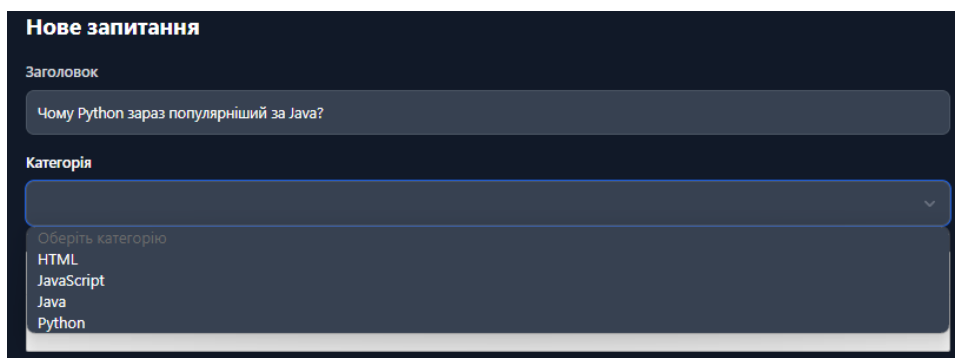
This image shows the same 'Нове запитання' form as in Figure 4.19, but with example data. The 'Заголовок' field contains the text 'Чому Python зараз популярніший за Java?'. The 'Категорія' dropdown menu is open, showing a list of categories: 'HTML', 'JavaScript', 'Java', and 'Python'. The 'Python' option is highlighted.

Рисунок 4.20 – Приклад заповненої форми для створення запитання

*Джерело:* знімок з екрану [40]

При введенні некоректної інформації чи не заповненні обов'язкових полів форми, користувач отримає нотифікацію про помилку (рис.4.21). Аналогічно до успішного створення запитання, користувач отримає повідомлення про збереження даних (рис.4.22).

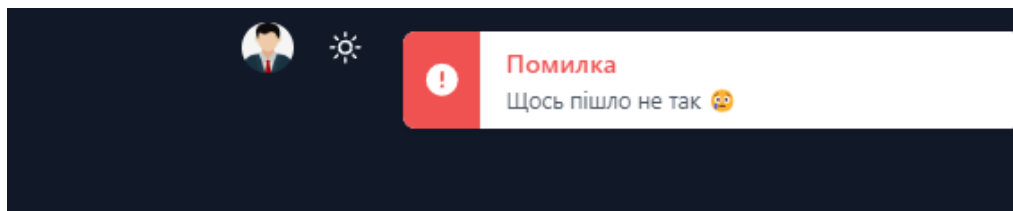


Рисунок 4.21 – Нотифікація про помилку при створенні запитання  
*Джерело:* знімок з екрану [40]

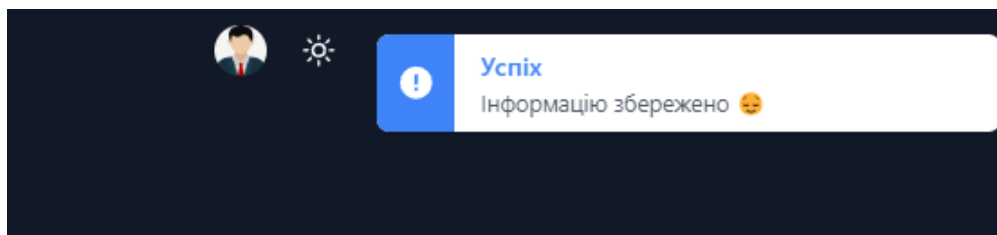


Рисунок 4.22 – Нотифікація про збереження запитання  
*Джерело:* знімок з екрану [40]

Перейшовши на сторінку створеного запитання, користувач має можливість отримати відповідь сформованою ChatGPT. Для цього у користувача є функціонал у вигляді кнопки «Запитати в ChatGPT», що знаходиться під детальним описом запитання (рис.4.23).



Рисунок 4.23 – Сформоване запитання користувача  
*Джерело:* знімок з екрану [40]

Після натискання на кнопку відбувається запит до API ChatGPT. Користувачу відображається анімація завантаження інформації (рис.4.25).

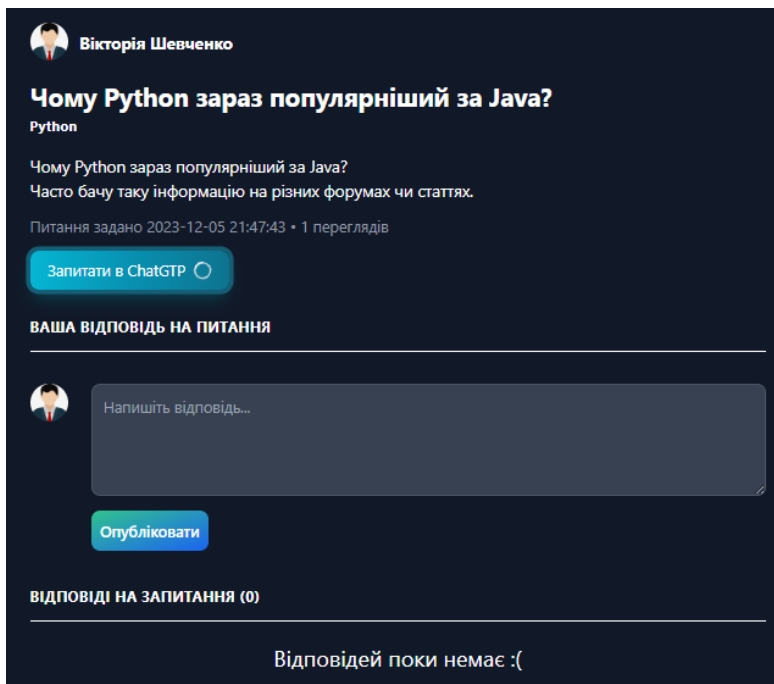


Рисунок 4.24 – Завантаження відповіді

*Джерело:* знімок з екрану [40]

Після отримання відповіді від API ChatGPT користувач має можливість опублікувати її, у разі якщо вона є коректною та розкриває певні аспекти питання (рис.4.25). Так як була використана безкоштовна API ChatGPT, отримати повноцінну відповідь неможливо.

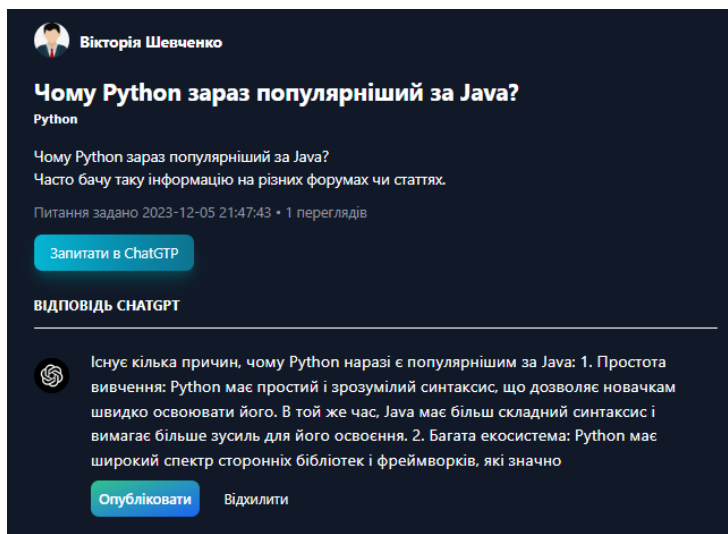


Рисунок 4.25 – Функція для публікації відповіді

*Джерело:* знімок з екрану [40]

Після публікації відповіді (рис.4.26), вона відображається саме як від користувача, у даному випадку ChatGPT.

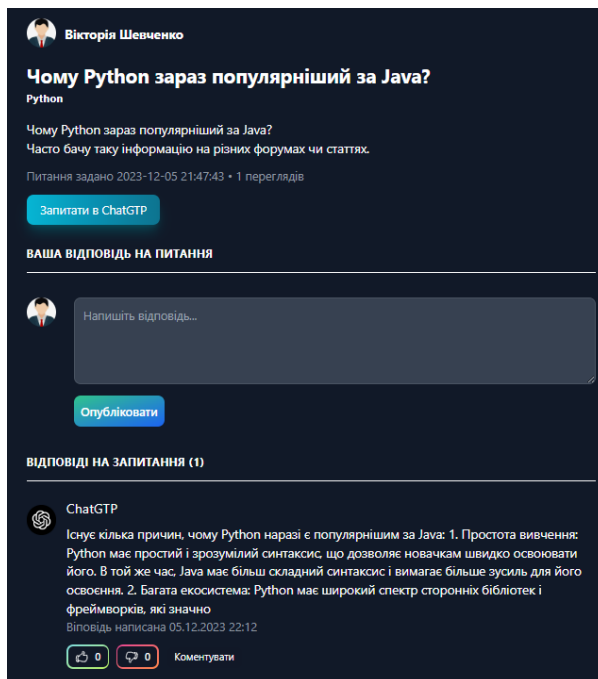


Рисунок 4.26 – Публікування відповіді API ChatGPT

*Джерело:* знімок з екрану [40]

Розглянемо детальніше сторінку із користувачами (рис.4.27). На даній сторінці відображаються усі користувачі та перелік самих активних користувачів web-орієнтованої системи.

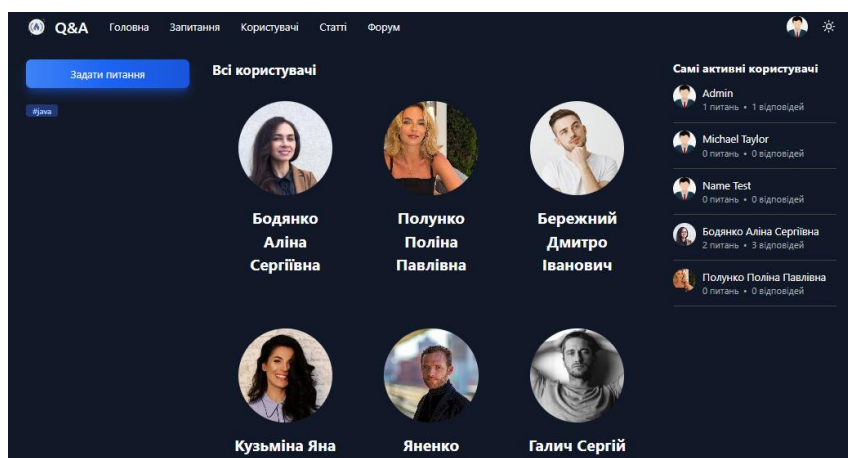


Рисунок 4.27 – Сторінка із користувачами

*Джерело:* знімок з екрану [40]

Перейшовши на сторінку обраного користувача, можемо переглянути детальну інформацію та його активність, а саме створенні запитання (рис.4.28), відповіді на запитання інших користувачів (рис.4.29), вподобання (рис.4.30) та створені публікації (рис.4.31).

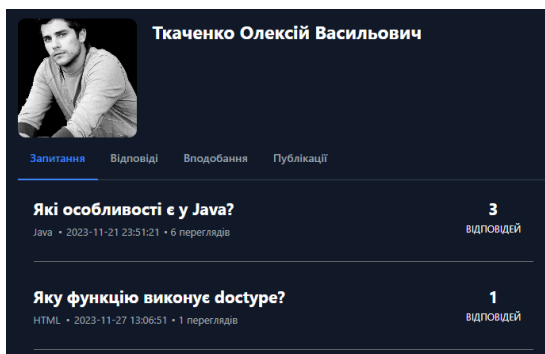


Рисунок 4.28 – Створені запитання користувача

*Джерело:* знімок з екрану [40]



Рисунок 4.29 – Надані відповіді на запитання інших користувачів

*Джерело:* знімок з екрану [40]

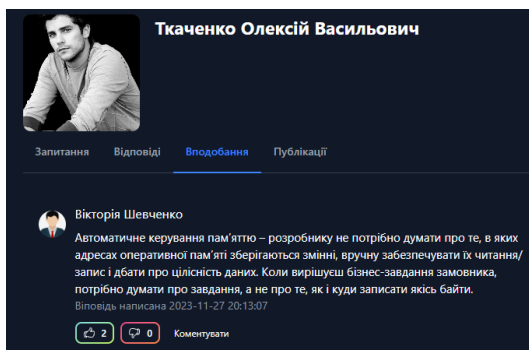


Рисунок 4.30 – Вподобання користувача

*Джерело:* знімок з екрану [40]

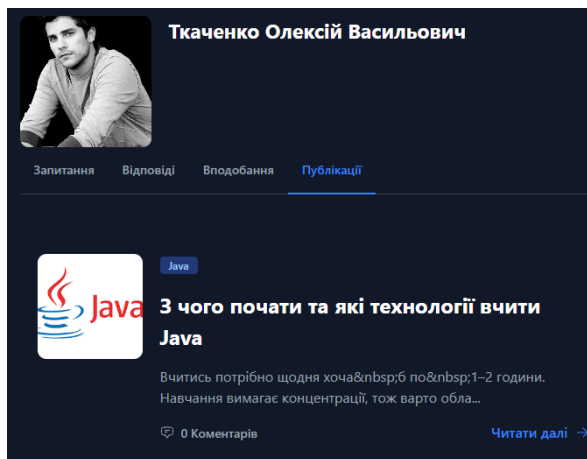


Рисунок 4.31 – Створені статі користувача

*Джерело:* знімок з екрану [40]

Перейдемо до сторінки із статтями користувачів web-орієнтованої системи (рис.4.32). На сторінці присутня фільтрація статей за популярністю та авторами. Також є пошук за назвою статі.

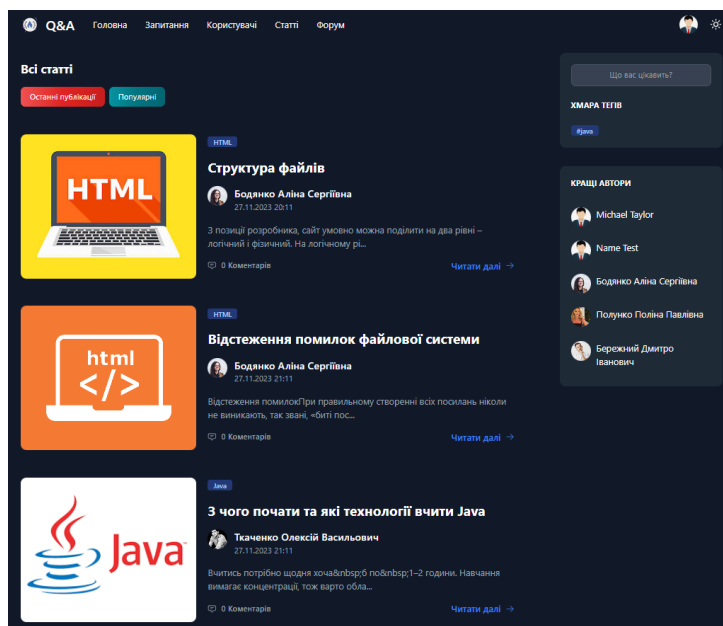


Рисунок 4.32 – Сторінка із статтями користувачів

*Джерело:* знімок з екрану [40]

Після пошуку, перейшовки на обрану статтю, можна переглянути детальну інформацію про неї (рис.4.33). Також на сторінці відображається інформація про інші популярні статі web-орієнтованої системи.



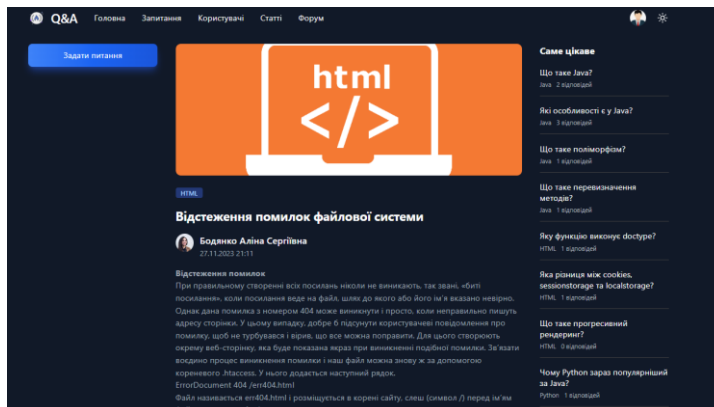


Рисунок 4.33 – Детальна інформація обраної статі

*Джерело:* знімок з екрану [40]

В кожній статі користувач має можливість залишити коментар. Для цього необхідно заповнити форму та опублікувати коментар. Приклад коментарю на рисунку 4.34.

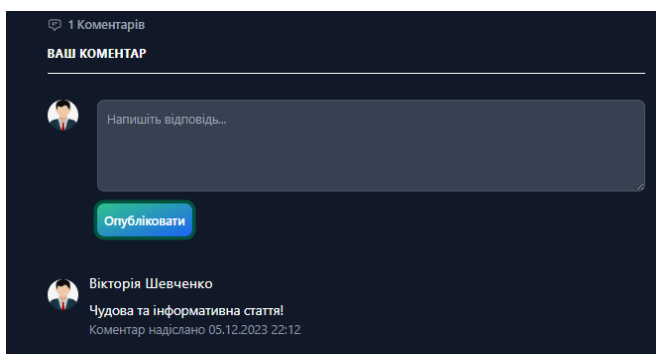


Рисунок 4.34 – Приклад коментарю в web-орієнтованій системі

*Джерело:* знімок з екрану [40]

Кожний зареєстрований користувач web-орієнтованої системи має можливість написати та опублікувати власну статтю із корисною інформацією. Щоб використати даний функціонал необхідно натиснути на профіль, а потім на поле «Статті» (рис.4.35).

Після натискання, користувач перенаправляється на сторінку із переліком власних статей. У прикладі, на рисунку 4.36, у користувача ще не створено жодної статі.

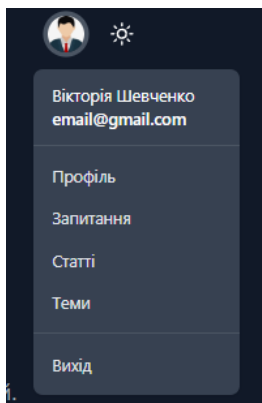


Рисунок 4.35 – Відображення функціоналу в профілі

*Джерело:* знімок з екрану [40]

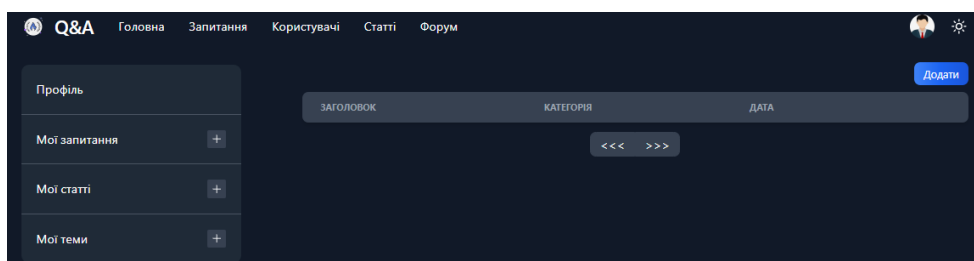


Рисунок 4.36 – Сторінка із статтями користувача

*Джерело:* знімок з екрану [40]

Після натискання кнопки «Додати» користувач переходить на сторінку створення статі. Необхідно заповнити всю необхідну інформацію та додати картинку для обкладинки (рис.4.37).

Рисунок 4.37 – Форма для додавання статі

*Джерело:* знімок з екрану [40]

Після створення нової статті, вона відображається у списку на сторінці користувача (рис.4.38). Перейшовши за посиланням на статтю, можна переглянути детальну інформацію, як на рисунку 4.39. Вже створену статтю можна видаляти та редагувати. Після збереження змін, під час редагування, користувач отримує нотифікацію про збереження даних (рис.4.40).

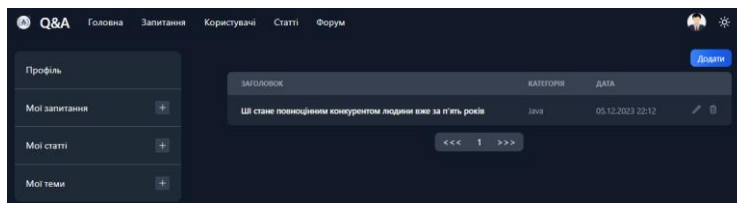


Рисунок 4.38 – Збережена нова стаття

*Джерело: знімок з екрану [40]*

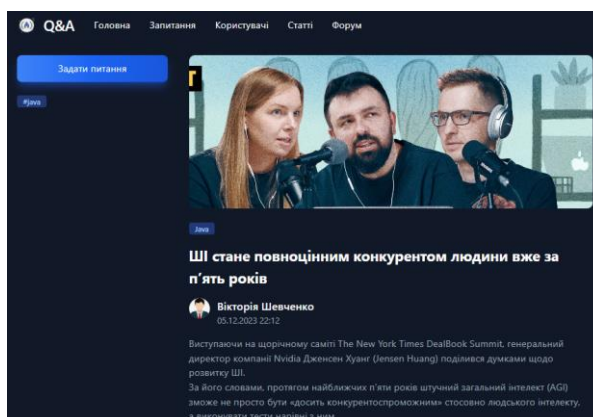


Рисунок 4.39 – Відображення статі при перегляді детальної інформації

*Джерело: знімок з екрану [40]*

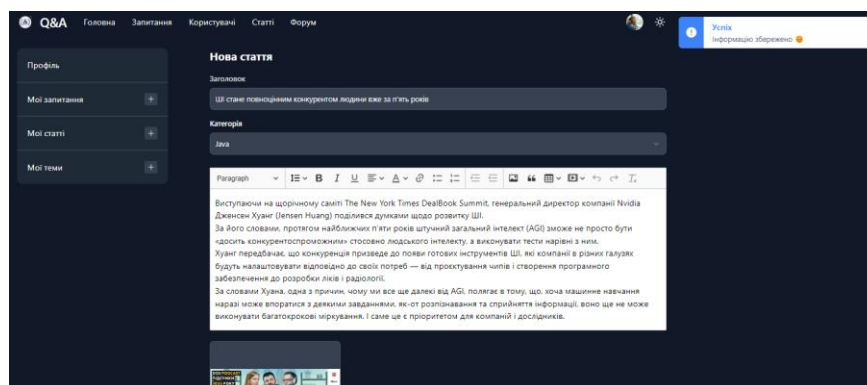


Рисунок 4.40 – Збереження відредагованої статі

*Джерело: знімок з екрану [40]*

Перейдемо до сторінки із форумом (рис.4.41). На даній сторінці реалізований функціонал для обговорення тематичних питань.

Приклад обговорення представлено на рисунку 4.42.

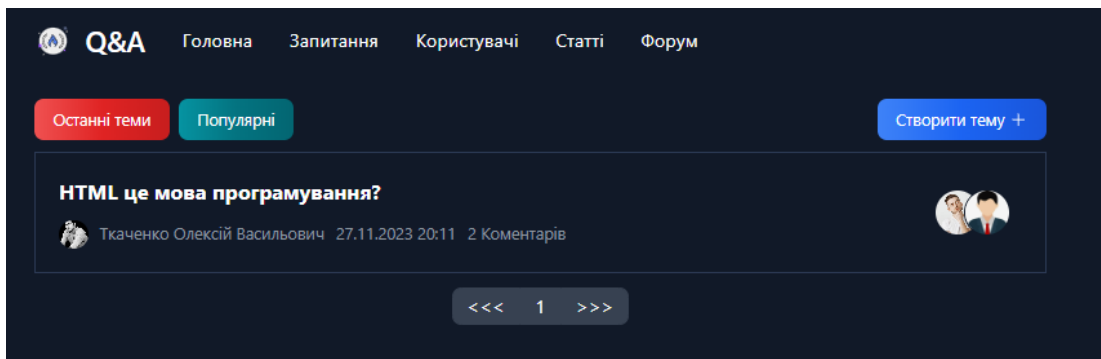


Рисунок 4.41 – Сторінка форуму

*Джерело:* знімок з екрану [40]

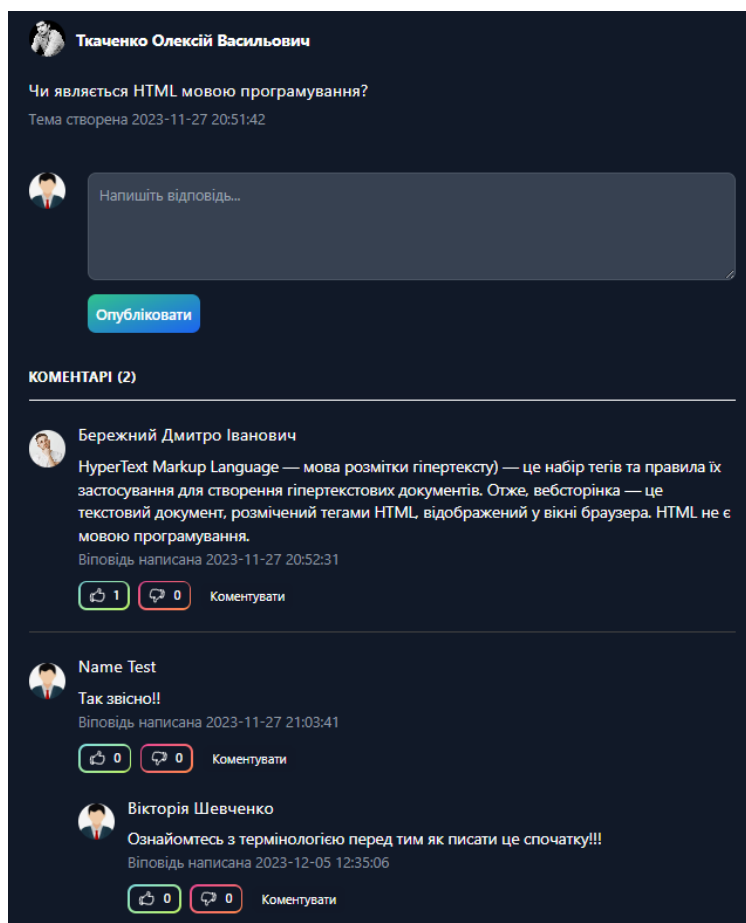


Рисунок 4.42 – Приклад обговорення

*Джерело:* знімок з екрану [40]

При обговоренні та додаванні коментарів не можна надсилати повідомлення з певним переліком слів, що додаються з адміністративної панель. При введенні таких слів користувач отримає відповідну нотифікацію, як на рисунку 4.43.

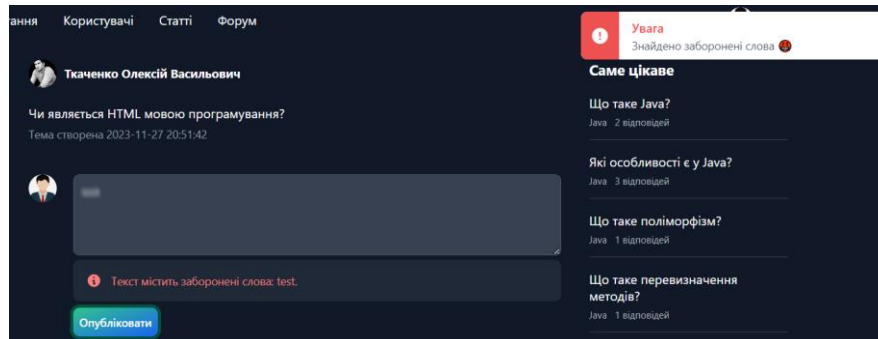


Рисунок 4.43 – Нотифікація про введення заборонених слів

*Джерело:* знімок з екрану [40]

Також кожен коментар можна оцінити позитивно та негативно. Приклад позитивного оцінювання коментарю представлено на рисунку 4.44 та негативного на рисунку 4.45.

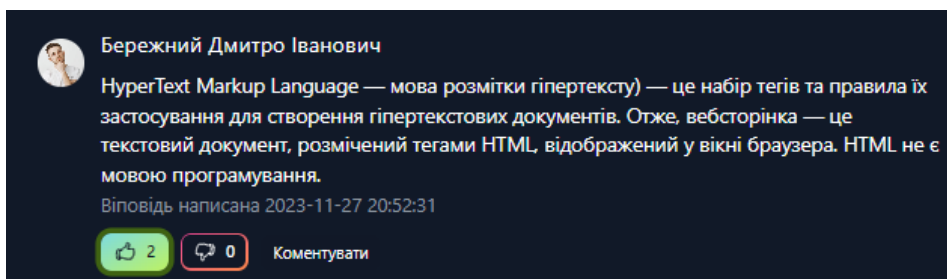


Рисунок 4.44 – Позитивне оцінювання

*Джерело:* знімок з екрану [40]

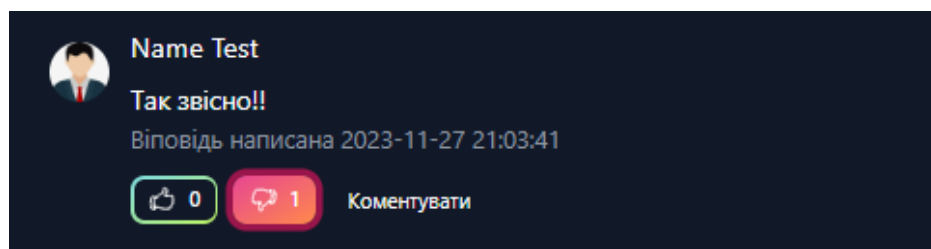


Рисунок 4.45 – Негативне оцінювання

*Джерело:* знімок з екрану [40]

Також кожен користувач може створити власну тему для обговорення на сторінці користувача (рис.4.46).

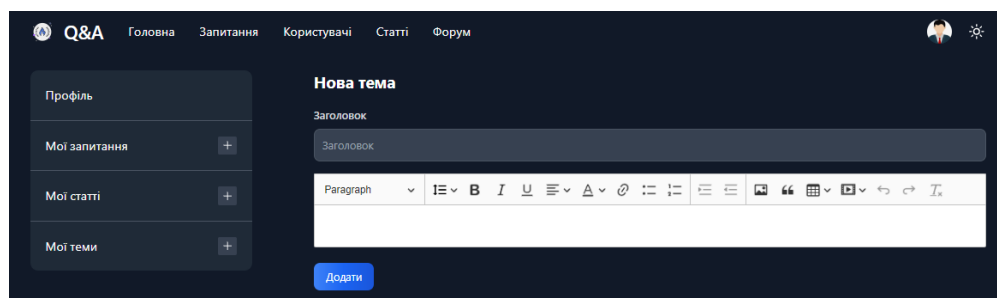


Рисунок 4.46 – Додавання обговорення

*Джерело:* знімок з екрану [40]

Також можна оновлювати фото профілю, що також буде відображатися для всіх користувачів (рис 4.47-48). При редагуванні особистої сторінки можна також додати інформацію про себе та посилання на соціальні мережі. Ці посилання будуть відображатися для інших користувачів на вашій особистій сторінці (рис.4.49).

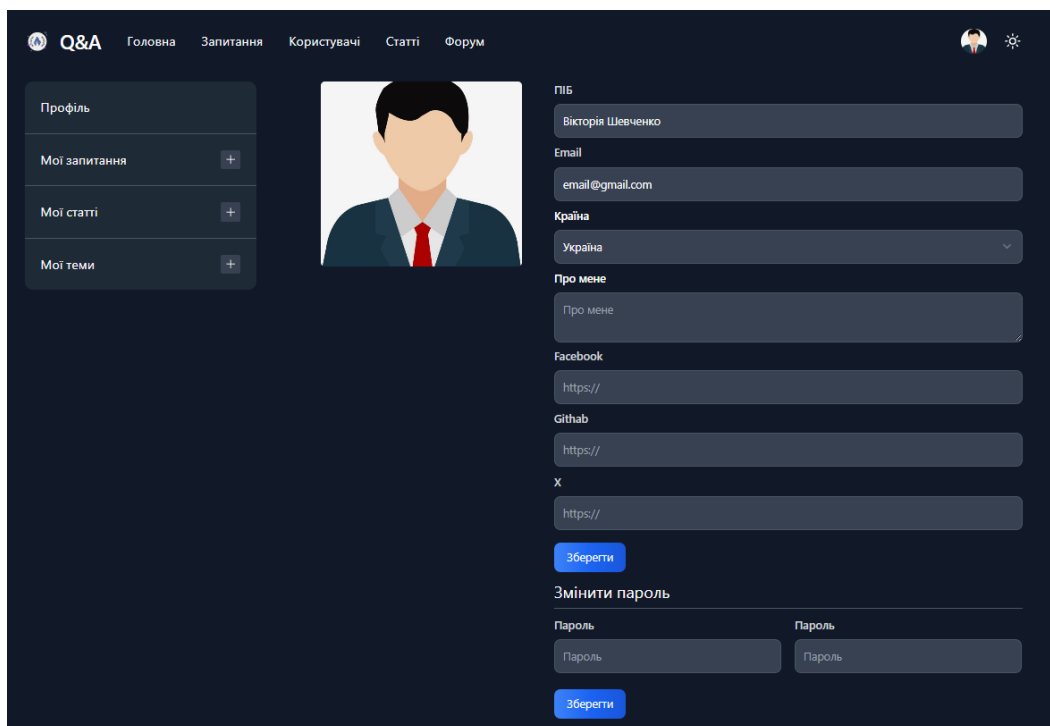


Рисунок 4.47 – Редагування особистої сторінки

*Джерело:* знімок з екрану [40]



Рисунок 4.48 – Відредагована сторінка користувача

*Джерело:* знімок з екрану [40]

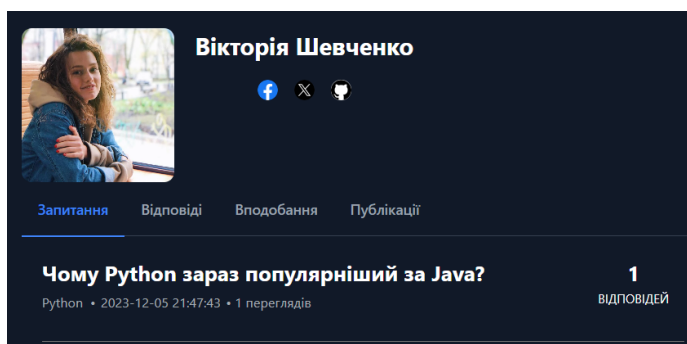


Рисунок 4.49 – Посилання на сторінки

*Джерело:* знімок з екрану [40]

### 4.3 Адміністрування web-орієнтованої системи

Перейшовши на адміністративну панель, можна переглянути зареєстрованих користувачів та заблокувати у разі потреби (рис.4.50).

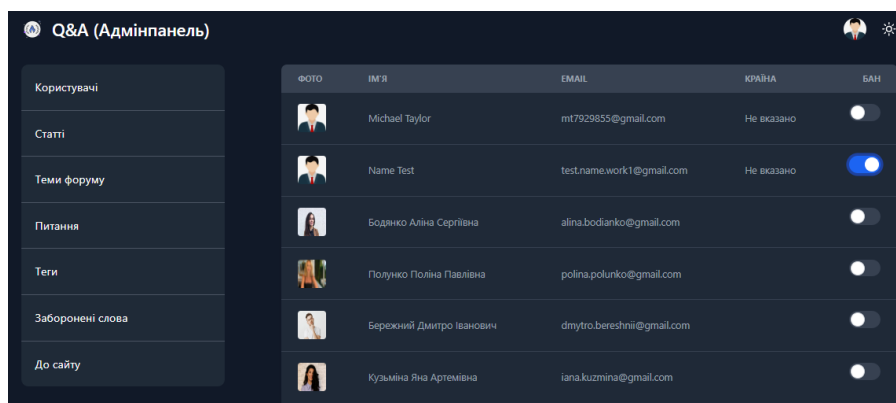
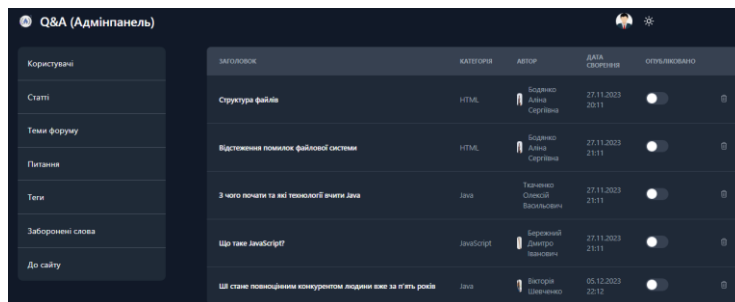


Рисунок 4.50 – Перелік користувачів

*Джерело:* знімок з екрану [40]

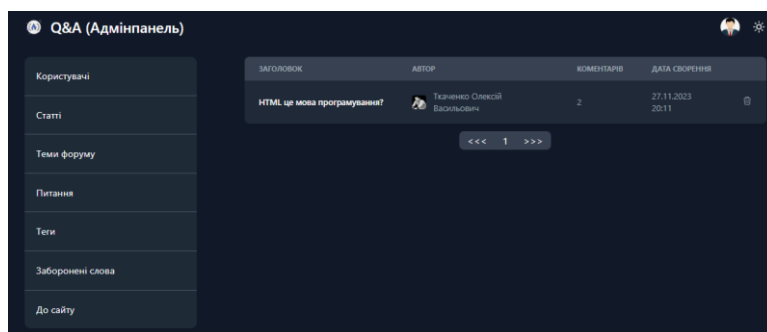
Крім цього користувач може переглядати та блокувати створені статі та обговорення користувачів системи (рис.4.51-52).



Користувачі	ЗАГОЛОВОК	КАТЕГОРІЯ	АВТОР	ДАТА СВЕРНЕННЯ	ОПЬЛЮВАНЬО
Статті	Структура файла	HTML	Бодяно Аліна Сергіївна	27.11.2023 20:11	🔍
Теми форуму	Відставлення помилок файлової системи	HTML	Бодяно Аліна Сергіївна	27.11.2023 21:11	🔍
Питання	З чого почати та які технології вчити Java	Java	Ткаченко Олександр Васильович	27.11.2023 21:11	🔍
Теги	Що таке Lombok?	Lombok	Бережний Дмитро Іванович	27.11.2023 21:11	🔍
Заборонені слова	Що стане повноцінним конкурентом людям вже за 7 років?	Java	Ткаченко Олександр Васильович	05.12.2023 22:12	🔍
До сайту					

Рисунок 4.51 – Перелік Категорій

*Джерело:* знімок з екрану [40]

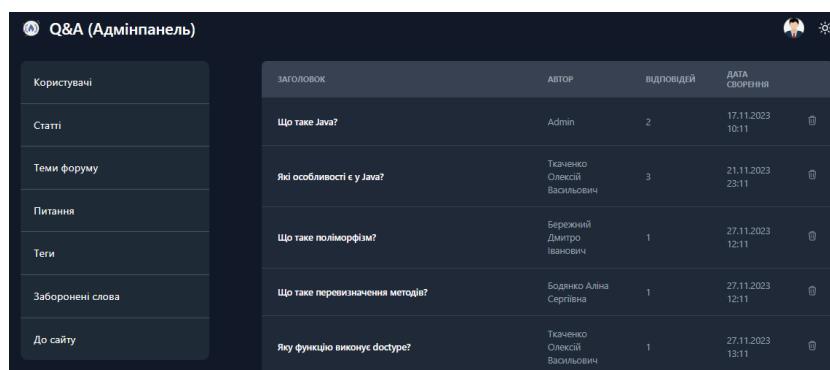


Користувачі	ЗАГОЛОВОК	АВТОР	КОМЕНТАРІВ	ДАТА СВЕРНЕННЯ
Статті	HTML це мова програмування?	Ткаченко Олександр Васильович	2	27.11.2023 20:11
Теми форуму				
Питання				
Теги				
Заборонені слова				
До сайту				

Рисунок 4.52 – Перелік обговорень

*Джерело:* знімок з екрану [40]

Адміністратор може переглядати також запитання створені в системі та приймати участь у обговореннях (рис.4.53).



Користувачі	ЗАГОЛОВОК	АВТОР	ВІДПОВІДЕЙ	ДАТА СВЕРНЕННЯ
Статті	Що таке Java?	Admin	2	17.11.2023 10:11
Теми форуму	Які особливості є у Java?	Ткаченко Олександр Васильович	3	21.11.2023 23:11
Питання	Що таке поліморфізм?	Бережний Дмитро Іванович	1	27.11.2023 12:11
Теги	Що таке порівняння методів?	Бодяно Аліна Сергіївна	1	27.11.2023 12:11
Заборонені слова	Яку функцію виконує доступ?	Ткаченко Олександр Васильович	1	27.11.2023 13:11
До сайту				

Рисунок 4.53 – Запитання користувачів

*Джерело:* знімок з екрану [40]



Також присутня можливість видаляти створені хештеги користувачів (рис.4.54). Для введення обмежень на слова при написанні коментарів створено окрему сторінку в адміністративній панелі (рис.4.55).

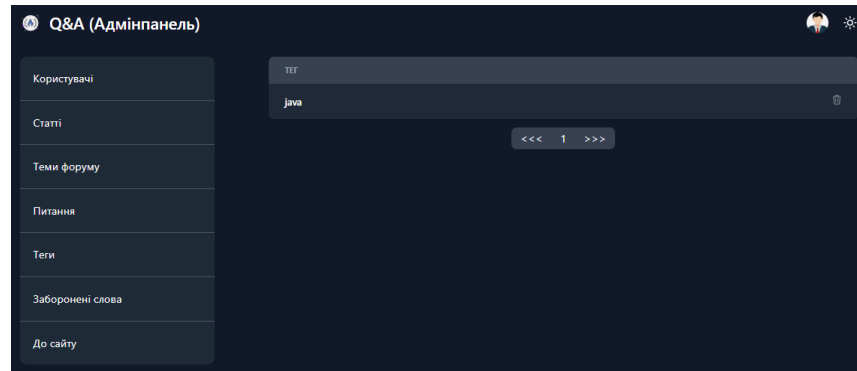


Рисунок 4.54 – Перелік хештегів

*Джерело:* знімок з екрану [40]

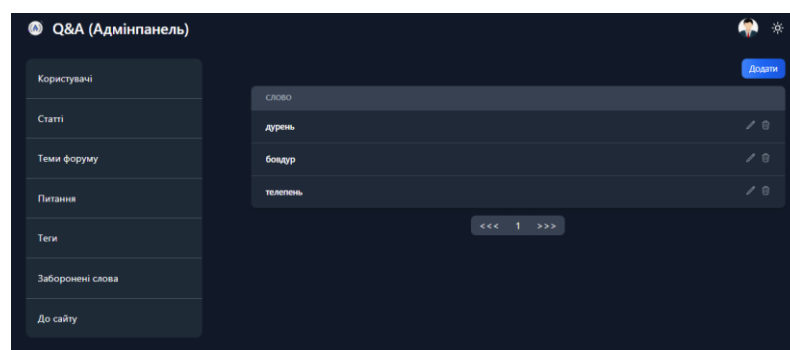


Рисунок 4.55 – Перелік заборонених слів

*Джерело:* знімок з екрану [40]

#### 4.4 Тестування та дослідження працездатності web-орієнтованої системи

Для перевірки працездатності web-орієнтованої системи консультування, які були реалізовані, необхідно повноцінне тестування. Це включає в себе авторизацію, реєстрацію, сторінку форуму, сторінку із запитаннями та інше. Результати проведення тестування представлено в таблиці 4.1.

Таблиця 4.1 – Тестування web-орієнтованої системи консультування

№	Назва тесту	Очікуваний результат	Фактичний результат	0/1
1. Сторінка реєстрації				
1.1	Перевірка форми реєстрації при введенні всіх даних	Успішна реєстрація користувача.	Після реєстрації користувач перенаправляється на сторінку на особисту сторінку для редагування даних.	1
1.2	Перевірка форми реєстрації при введенні лише електронної пошти чи ім'я	Нотифікація про відсутність обов'язкових для введення даних.	Користувач отримує нотифікацію «Заповніть це поле», у відповідному пропущеному полі форми реєстрації	1
1.3	Перевірка форми реєстрації при використанні Google авторизації	Успішна реєстрація користувача з використанням даних Google.	Переадресація на сторінку вибору Google-акаунту для використання даних.	1
1.4	Перевірка форми реєстрації при використанні GitHub авторизації	Успішна реєстрація користувача з використанням даних GitHub.	Переадресація на сторінку авторизації сервісу GitHub для використання даних.	1
2. Сторінка авторизації				
2.1	Перевірка форми авторизації при введенні всіх даних	Успішна авторизація користувача.	Переадресація на головну сторінку з використанням авторизованого користувача	1

Продовження таблиці 4.1

№	Назва тесту	Очікуваний результат	Фактичний результат	0/1
2.2	Перевірка форми авторизації при введенні лише електронної пошти або паролю	Відмова в авторизації користувача.	Нотифікація про помилку авторизації користувача із-за введення неправильних даних.	1
2.3	Перевірка форми авторизації при використанні Google аккаунту	Успішна авторизація користувача з використанням даних Google.	Переадресація на сторінку вибору Google-аккаунту для використання даних.	1
2.4	Перевірка форми авторизації при використанні GitHub аккаунту	Успішна авторизація користувача з використанням даних GitHub.	Переадресація на сторінку авторизації сервісу GitHub для використання даних.	1
3. Сторінка користувачів				
3.1	Натискання на будь-якого користувача	Відкриття сторінки вибраного користувача.	Переадресація на сторінку з користувачем де надані короткі відомості про запитання, відповіді, вподобання та публікації обраного користувача.	1
3.2	Вибір користувача через список найактивніших користувачів	Відкриття сторінки вибраного користувача.	Переадресація на сторінку з користувачем де надані короткі відомості про запитання, відповіді, вподобання та публікації обраного користувача.	1

Продовження таблиці 4.1

№	Назва тесту	Очікуваний результат	Фактичний результат	0/1
3.3	Натискання на «Задати запитання»	Відкриття сторінки з можливістю задати питання.	Переадресація на сторінку з можливістю задати нове питання, детально відредагувати введений опис та присвоїти йому категорію.	1
3.4	Вибір хеш-тегу	Показ списку питань з обраним хеш-тегом.	Переадресація на сторінку зі списком питань під обраний хеш-тег. Також користувач може фільтрувати список питань за категоріями, популярністю, датою створення та наявністю відповіді.  Присутня можливість конкретного питання та створення нового.	1
<b>4. Сторінка запитань</b>				
4.1	Натискання на «Задати запитання»	Відкриття сторінки з можливістю задати питання.	Переадресація на сторінку з можливістю задати нове питання, детально відредагувати введений опис та присвоїти йому категорію.	1

Продовження таблиці 4.1

№	Назва тесту	Очікуваний результат	Фактичний результат	0/1
4.2	Вибір хеш-тегу	Показ списку питань з обраним хеш-тегом.	До наявного сортованого та фільтрованого списку питань додається фільт за хеш-тегом.	1
4.3	Натискання на «Пошук»	Показ списку питань за обраними фільтрами.	Застосування обраних фільтрів та використання поля для конкретного пошуку потрібного питання.	1
4.4	Фільтрація питань за категоріями	Представлення списку за обраною категорією.	Застосування фільтру «Категорія» для фільтрації питань.	1
4.5	Сортування питань за датою створення, наявністю відповіді та популярністю	Представлення списку найсвіжіших питань, питань без відповідей або найпопулярніших запитань.	Застосування сортування питань для показу останніх створених питань, питань на які ще не надано відповіді, або найпопулярніші питання серед користувачів.	1
4.6	Вибір потрібного питання	Відкриття сторінки з питанням.	Переадресація на сторінку з обраним питанням, надання опису, відомостей та наявність відповіді до нього. Також представлена інформація про автора питання, обговорення між користувачами.	1

## Продовження таблиці 4.1

№	Назва тесту	Очікуваний результат	Фактичний результат	0/1
4.7	Вибір питання зі списку «Саме цікаве»	Відкриття сторінки з питанням.	Переадресація на сторінку з обраним питанням, надання опису, відомостей та наявність відповіді до нього. Також представлена інформація про автора питання, обговорення між користувачами про правильність відповіді та рейтинг тієї чи іншої відповіді.	1
<b>5. Сторінка статей</b>				
5.1	Сортування статей за датою створення та популярністю	Показ списку статей за новизною або популярністю.	Застосування сортування до списку статей за новизною створення та популярністю серед користувачів.	1
5.2	Пошук конкретної статті	Представлення статті за заданим пошуком.	Застосування фільтрування статей за ключовими словами, що введені в полі пошуку статей	1
5.3	Фільтрація статей за хеш-тегами	Показ списку статей з обраним хеш-тегом.	Застосування фільтру для відображення статей за обраним хеш-тегом.	1

Продовження таблиці 4.1

№	Назва тесту	Очікуваний результат	Фактичний результат	0/1
5.4	Вибір користувача зі списку «Найкращі автори»	Відкриття сторінки вибраного користувача.	Переадресація на сторінку з користувачем де надані короткі відомості про запитання, відповіді, вподобання та публікації обраного користувача.	1
5.5	Натискання на автора потрібної статті	Відкриття сторінки вибраного користувача.	Переадресація на сторінку з користувачем де надані короткі відомості про запитання, відповіді, вподобання та публікації обраного користувача.	1
5.6	Вибір потрібної статті	Показ статті та опису до неї.	Переадресація на сторінку статті з детальним її описом та коментарями від інших користувачів. Вказується автор статті та короткі відомості про статтю. Також користувач може переглянути список найцікавіших питань або задати власне.	1
5.7	Перегляд коментарів потрібної статті	Показ коментарів обраної статті.	Переадресація на сторінку статті та автоматичний скорл до коментарів обраної статті.	1

Продовження таблиці 4.1

№	Назва тесту	Очікуваний результат	Фактичний результат	0/1
6. Сторінка форуму				
6.1	Сортування тем форуму за датою створення та популярністю	Показ списку тем за новизною або популярністю.	Застосування сортування до списку тем за новизною створення та популярністю серед користувачів.	1
6.2	Створення нової теми	Відкриття сторінки з можливістю створити нову тему для форуму.	Переадресація на сторінку з можливістю задати нову тему форуму, детально відредагувати введений опис та присвоїти їй категорію.	1
6.3	Пошук конкретної теми	Представлення тем за заданим пошуком.	Застосування фільтрування тем за ключовими словами, що введені в полі пошуку тем	1
6.4	Фільтрація тем за хеш-тегами	Показ списку тем за обраним хеш-тегом.	Застосування фільтру для відображення тем за обраним хеш-тегом.	1
6.5	Вибір потрібної теми	Представлення теми та коментарів до неї.	Переадресація на сторінку теми форуму з детальним її описом та коментарями від інших користувачів. Вказується автор створеної теми та короткі відомості про неї.	1



Продовження таблиці 4.1

№	Назва тесту	Очікуваний результат	Фактичний результат	0/1
6.6	Вибір користувача зі списку «Найкращі автори»	Відкриття сторінки вибраного користувача.	Переадресація на сторінку з користувачем де надані короткі відомості про запитання, відповіді, вподобання та публікації обраного користувача.	1

*Джерело: побудовано автором*

Для викладення роботи на хостинг було обрано «Fornex» (рис.4.56) [41]. Першим етапом є реєстрація для використання функціонала.

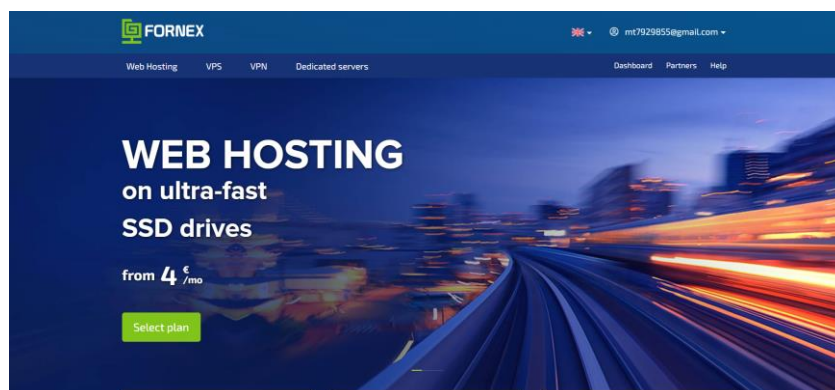


Рисунок 4.56 – Сторінка хостингу «Fornex»

*Джерело:* знімок з екрану [41]

Для викладення сайту на хостинг потрібно обрати тарифний план, що включає в себе певні обмеження в залежності від ціни (рис.4.57). Для web-орієнтованої системи консультування було обрано мінімальний план.

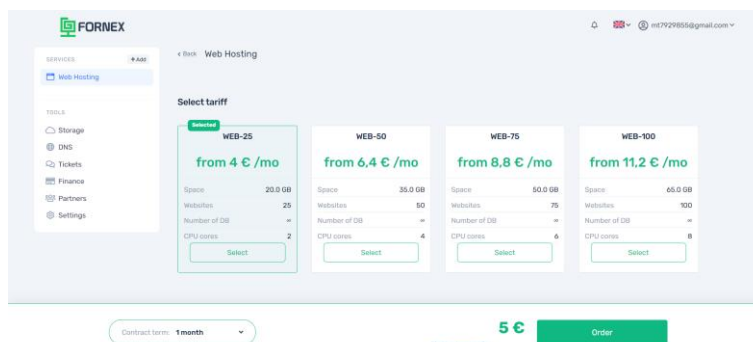


Рисунок 4.57 – Обрання тарифного плану

*Джерело:* знімок з екрану [41]

Крім того, попередньо потрібно виконати купівлю домену на будь-якому зручному сервісі. Адже наступний крок є створення сайту з доданням домену та завантаженням розробленого проєкту на хостинг (рис.4.48).

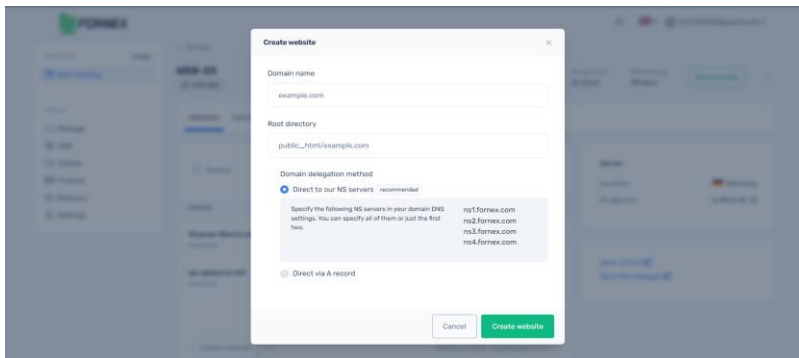


Рисунок 4.58 – Створення сайту на основі домену

*Джерело:* знімок з екрану [41]

Переходимо в файловий менеджер, переходимо в каталог public\_html/наш\_домен та натискаємо кнопку «Завантажити» (рис.4.59).

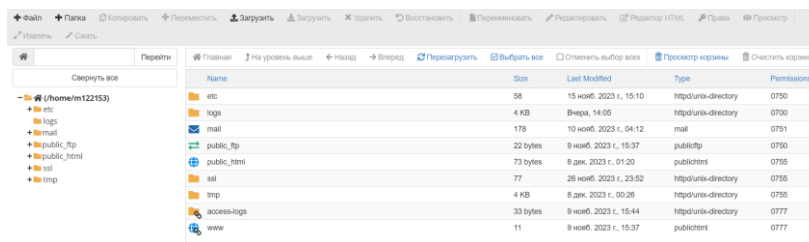


Рисунок 4.59 – Завантаження проекту

*Джерело:* знімок з екрану [41]

Для роботи із базою даних буде використано вже створену попередньо базу даних. Для цього надається доступ до інтерфейсу phpMyAdmin (рис.4.60).

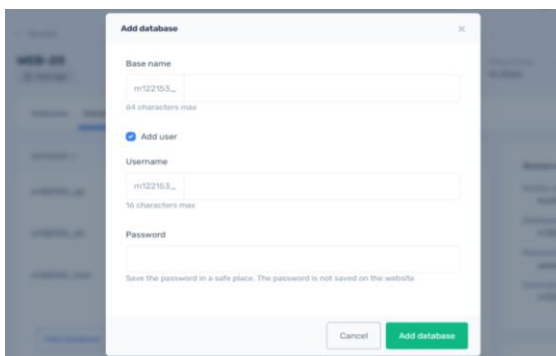


Рисунок 4.60 – Завантаження проекту

*Джерело:* знімок з екрану [41]

Наступний крок – завантажуюмо нашу базу даних на хостинг (рис.4.61). Також на відповідній сторінці обираємо актуальну версію PHP.

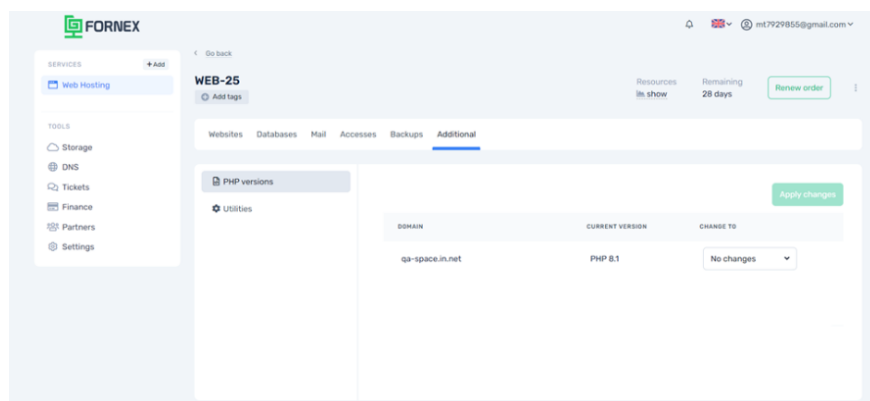


Рисунок 4.61 – Встановлення актуальну версію PHP

*Джерело:* знімок з екрану [41]

Отже, web-орієнтованої система консультування була викладена на хостинг та вірно налаштована. У результаті проведеного тестування дефектів не було виявлено. Увесь функціонал web-орієнтованої системи консультування працює коректно.

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи магістра на тему "Web-орієнтована система консультування IT-фахівців з професійних питань " було створено web-орієнтовану систему у вигляді web-додатку під назвою «FinanceFlow».

Провівши ретельний аналіз предметної області та дослідження аналогів-конкурентів, було встановлено базовий функціонал та отримано візуальне представлення вигляду системи, а також підтверджено актуальність проекту. Створено планування робіт у вигляді календарного плану з візуалізацією діаграми Ганта та проведено оцінку можливих ризиків при створенні проекту. Для створення web-орієнтованої системи було підібрано оптимальне програмне забезпечення та підібрано найкращий штучний інтелект під обрані задачі.

Після постановки мети та формулювання задач проекту було розроблено візуальну та функціональну частину web-орієнтованої системи. Спроектовано функціонал системи у вигляді схем та представлено схеми варіантів використання web-орієнтованої системи користувачем та адміністратором. Створено базу даних з логічними зв'язками між блоками та компонентами.

Рободу web-орієнтованої системи представлено у вигляді візуального використання як користувачем, так і адміністратором. Деведено велику користь від відповідей штучного інтелекта.

Web-орієнтована система була заповнена інформацією, а ретельне тестування допомогло усунути помилки в роботі системи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The best technical articles from IT professionals of 2021 [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://dou.ua/lenta/articles/best-tech-blogs-2021/> (дата звернення: 04.10.2023).
2. The real benefit of forums [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <http://it-forum.com.ua/it-forum/> (дата звернення: 04.10.2023).
3. Communication in forums [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://replace.org.ua/topic/> (дата звернення: 06.10.2023).
4. Lakshmi, Y.H., Sangeetha, Y., Sri Vyshnavi, K.P., & Fyzulla, S. (2023). A Website for a Consultancy using Mern Stack. 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), 195-200.
5. Are forums really that useful? [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://replace.org.ua/topics/719/> (дата звернення: 12.10.2023).
6. Sin, K.F. (2009). Implementation of web 2.0 in training consultancy industry.
7. Double bottom on the topic of the forums [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://qna.habr.com/q/178301> (дата звернення: 12.10.2023).
8. Everything you wanted to know about the Forum [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://bookforum.ua/uk/p/vse-hotily-znaty-pro-forum/> (дата звернення: 12.10.2023).
9. An Introduction to Data Encoding and Decoding in Data Science [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.sitepoint.com/data-encoding-decoding-data-science> (дата звернення: 12.10.2023).
10. Leverage GPT-4 for Debugging and Bug Fixes [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.sitepoint.com/gpt-for-debugging/> (дата звернення: 15.10.2023).

11. No-code AI: The Key to Unlocking Efficiency and Growth [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.sitepoint.com/no-code-ai-the-key-to-unlocking-efficiency-and-growth/> (дата звернення: 15.10.2023).

12. Stack Overflow will add AI-generated answers to search results [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://ain.ua/2023/07/28/stack-overflow-own/> (дата звернення: 18.10.2023).

13. Do you use AI services on an ongoing basis? [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://ain.ua/2023/07/18/chy-vykorystovuyete-vy-ai-servisy-na-postijnij-osnovi/> (дата звернення: 18.10.2023).

14. And again about artificial intelligence. Help, threat or empty talk? [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://yur-gazeta.com/publications/practices/inshe/i-znovu-pro-shtuchniy-intelekt-dopomoga-zagroza-chi-pusti-balachki.html> (дата звернення: 19.10.2023).

15. Le-Tien, Thuong et al. “Developing a Chatbot system using Deep Learning based for Universities consultancy.” 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM) (2022): 1-7.

16. Baz, F.Ç., & Denizler, E. (2022). Web based remote agricultural control and consultancy application: An early diagnostic warning system. *European Mechanical Science*.

17. Ma, T., & Zhang, Z. (2021). Medical Consultation System based on Python Web crawler. 2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT), 737-741.

18. Liu, X., Xu, Z., Yu, X., & Oda, T. (2022). Why should I consult? The impact of social support on patient consultation in online healthcare communities. *Frontiers in Psychology*, 13.

19. Fan, W., Zhou, Q., Qiu, L., & Kumar, S. (2022). Should Doctors Open Online Consultation Services? An Empirical Investigation of Their Impact on Offline Appointments. *Information Systems Research*.

20. Feida, Chen. (2018). The legal consultation data and corpus of the thesis from China law network. Replication Data for: Design and research of legal consultation text classification system.

21. Yuspita, Y.E. (2023). Sistem Informasi Consultation Online Menggunakan Bahasa Pemrograman PHP dan Database MySQL. INCODING: Journal of Informatics and Computer Science Engineering.

22. Satria, D., Zulfan, Z., Munawir, M., & Mulyati, D.S. (2019). FINAL PROJECT CONSULTATION INFORMATION SYSTEM INTEGRATED NOTIFICATION SYSTEM BASED ON SMS GATEWAY. Cyberspace: Jurnal Pendidikan Teknologi Informasi.

23. Yuriska, Y. (2011). DESIGN OF POPULAR CONSULTATION (POLLING) Kampong Betawi USING PHP AND MACROMEDIA DREAMWEAVER MX.

24. Stack Overflow [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://stackoverflow.com/> (дата звернення: 20.10.2023).

25. Xu, B., Hoang, T., Sharma, A., Yang, C., Xia, X., & Lo, D. (2022). Post2Vec: Learning Distributed Representations of Stack Overflow Posts. IEEE Transactions on Software Engineering, 48, 3423-3441.

26. Slashdot [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://slashdot.org/> (дата звернення: 20.10.2023).

27. Sun, G., & Chen, C. (2021). Influence Maximization Algorithm Based on Reverse Reachable Set. Mathematical Problems in Engineering.

28. Code Project [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.codeproject.com/> (дата звернення: 20.10.2023).

29. Top Reasons to Choose Vue.js for App Development [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://radixweb.com/blog/reasons-to-choose-vuejs-for-app-development#Consider> (дата звернення: 28.11.2023).

30. Why We Choose Tailwind [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.pixel-plus.ch/post/why-we-choose-tailwind-for-frontend-development>. (дата звернення: 28.11.2023).



31. Tailwind CSS Components Tutorial [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.freecodecamp.org/news/tailwind-css-components-flowbite> (дата звернення: 18.10.2023).

32. Choose SASS [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://hackernoon.com/why-you-should-choose-sass> (дата звернення: 01.12.2023).

33. Benefits of using SASS [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.mugo.ca/Blogs/7-benefits-SASS-over-conventional-CSS> (дата звернення: 01.12.2023).

34. CKEditor 5 plugin [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://lebergssolutions.com/blog/how-integrates-ckeditor5-plugin-drupal-9-10-module> (дата звернення: 18.10.2023).

35. M. Rahmouni, M. Bouzaidi, S. Mbarki, Approach by modeling to generate an e-commerce web code from laravel. Indonesian Journal of Electrical Engineering and Computer Science. 30, 257–266 (2023).

36. M. Nugraha, R. Agus, H. Fathi, R. Baginda, DEVELOPMENT A WEB-BASED STUDENT INTERNSHIP APPLICATION USING LARAVEL FRAMEWORK. Journal of Information Technology and Its Utilization. 6, 31–38 (2023).

37. Designing for All: The Basic Principles of Web Accessibility [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.sitepoint.com/designing-for-all-the-basic-of-web-accessibility/> (дата звернення: 25.10.2023).

38. Corrado, E.M., & Frederick, K. (2008). Free and Open Source Options for Creating Database-Driven Subject Guides. Code4Lib Journal.

39. K. Seputra, L. Dewi, A Design of Patient Registration Apps Using Flutter, Laravel and, Vue JS. European Alliance for Innovation n.o., 2022.

40. Власний web-додаток «Q&A» [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://qa-space.in.net/articles> (дата звернення: 02.12.2023).

41. Web hosting [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://fornex.com/en/> (дата звернення: 10.12.2023).

## **ДОДАТОК А**

**ПЛАНУВАННЯ РОБІТ**  
**для розробки кваліфікаційної роботи магістра**  
**«Web-орієнтована система консультування ІТ фахівців з професійних**  
**питань»**

## А.1 ІДЕНТИФІКАЦІЯ МЕТИ ІТ-ПРОЕКТУ

Сутність деталізації мети проекту за допомогою SMART-методу впливає з розшифровки термінів, які формують його назву: конкретна (Specific), вимірювана (Measurable), досяжна (Achievable), реалістична (Relevant), обмежена у часі (Time-framed).

S – конкретність, специфічність. Вимагає щоб сформульована мета давала чітке якісне уявлення про специфічні унікальні та інноваційні властивості майбутнього продукту проекту порівняно з іншими його альтернативами.

M – вимірюваність. Передбачає показників вартості які вимірюються. При відсутності фізичних способів та інструментарію виміру використовуються експерти – як інструмент для виміру.

A – узгодженість. Встановлює, що мета повинна впливати з реальних проблем, місії, стратегічних планів, планів розвитку, а також узгоджуватись з інтересами зацікавлених сторін проекту.

R – реалістичність, релевантність. Показує, що мета є такою, яку можливо досягти з урахуванням реально доступних ресурсних можливостей та обмежень (людських, фінансових тощо).

Поставлена мета є досяжною, адже вона сформулювалася на основі реально доступних ресурсних можливостей та проведеного аналізу вже наявних досліджень експертів у даній сфері.

T – обмеженість в часі. Зумовлює необхідність «прив'язки» мети до певних обґрунтованих термінів її досягнення (або початку та тривалості дій по її досягненню).

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Мета web-орієнтованої системи є розробити якісний додаток для консультування ІТ-фахівців у сфері професійних питань, де штучний інтелект використовується як складова частина інформаційного середовища для покращення взаємодії між фахівцем і користувачем у різних аспектах..
Measurable (вимірювана)	Використовуючи мінімум ресурсів, розробити якісний програмний продукт.
Achievable (досяжна)	У наявності є всі необхідні технічні та програмні засоби.
Relevant (реалістична)	Розробники достатньо кваліфіковані для виконання поставлених задач.
Time-framed (обмежена у часі)	Ціль має часове обмеження. Терміни досягнення мети проекту визначаються порядком виконання кваліфікаційної роботи магістра.

Після проведення аналізу методом SMART можна визначити кінцеву мету: вчасно створити якісний програмний продукт з використанням мінімальних витрат.

## А.2 ПЛАНУВАННЯ ЗМІСТУ СТРУКТУРИ РОБІТ ІНФОРМАЦІЙНОЇ СИСТЕМИ

WBS – це графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов'язані з продуктом проекту. На верхньому першому рівні WBS фіксується продукт проекту. Він повинен відповідати продукту проекту. Наступний II рівень відповідає діям або основним заходам для досягнення продукту проекту. Потім триває розбивка цих дій доти, поки не відбувається виконання дій елементарних робіт.

Ієрархічна структура робіт являє собою, по суті, перелік завдань проекту. Вона може бути представлена в графічному вигляді або у вигляді опису, що відображає вкладення робіт. Ієрархічна структура робіт організовує і визначає весь зміст проекту. Роботи, не включені з WBS, не є роботами проекту.

Виконаємо побудову WBS структури, у якій зазначимо всі виконувані роботи в залежності від головних етапів:

Формування технічного завдання - розробка технічного завдання, що встановлює основне призначення, показники якості, техніко - економічні та спеціальні вимоги до розроблюваного інструментального засобу. Формування технічного завдання включає в себе підпункти:

- визначення предметної області;
- визначення мови написання;
- визначення цільової аудиторії;
- визначення вимог дизайну програмного продукту;
- визначення вимог засобів перегляду та вимог до системи управління контентом.

Планування проекту включає в себе розробку OBS структури, матриці відповідальності, календарного плану, а саме діаграми Ганта, управління ресурсами та ризиками.

Реалізація матиме 4 етапи:

Підготовка – збір потрібної інформації, формування можливостей та цілей проекту.

Проектування – розробка технічної моделі, створення ескізів основних об'єктів та ландшафту.

Розробка – створення програмного коду та графічної складової програми, тестування та оптимізація.

Реліз – випробування ПП, аналіз проблем та виправлення помилок.

І останній етап створення проекту завершення має на увазі здачу проекту в експлуатацію і закриття проекту.

Діаграму WBS наведено на рисунку А.1 та OBS на рисунку А.2.



Рисунок А.1 – Структура WBS

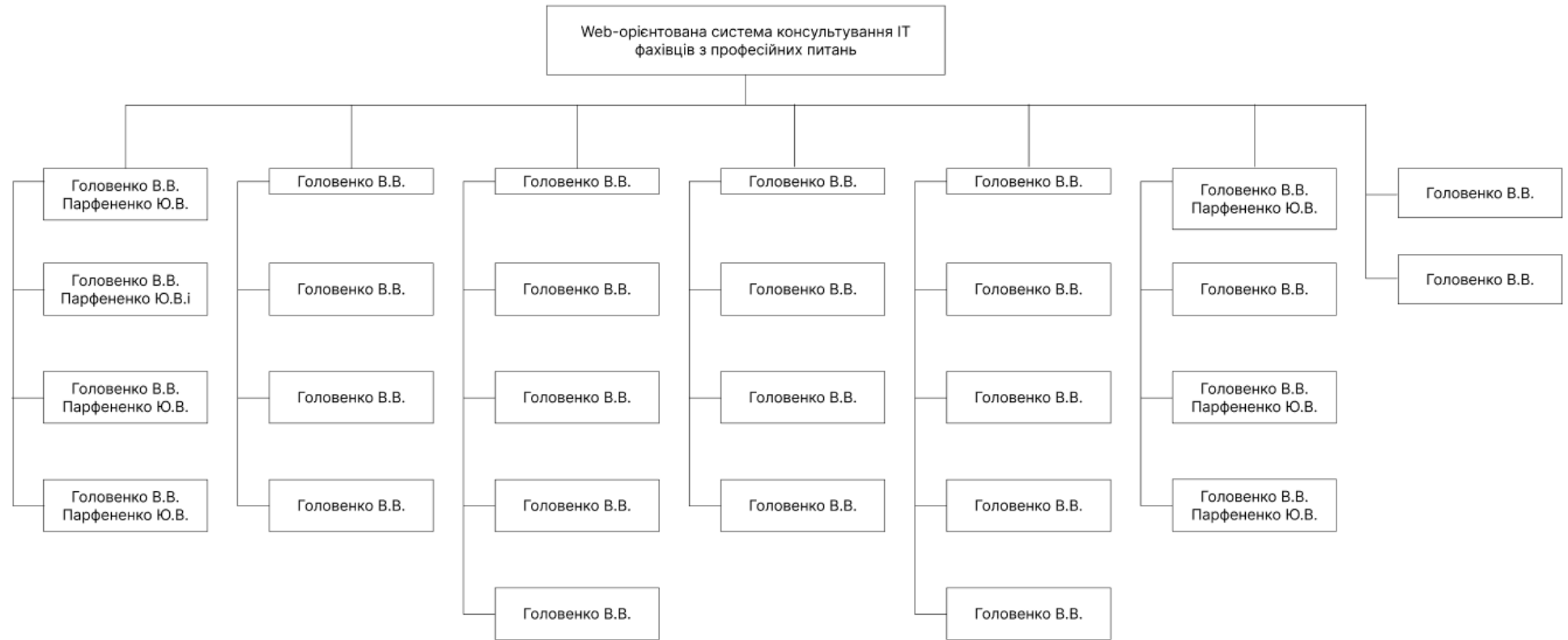


Рисунок А.2 – Структура OBS



### **А.3 ПОБУДОВА КАЛЕНДАРНОГО ГРАФІКУ ВИКОНАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ**

Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткової мережевої моделі будують календарний графік робіт.

Діаграма Ганта – горизонтальна лінійна діаграма, на якій задачі проекту представляються протяжними в часі відрізками, що характеризуються датами початку та закінчення, затримками і, можливо, іншими тимчасовими параметрами.

Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

На наступному рисунку представлено діаграму Ганта розроблюваного проекту. На рисунку А.3 представлено побудовану діаграму Ганта.

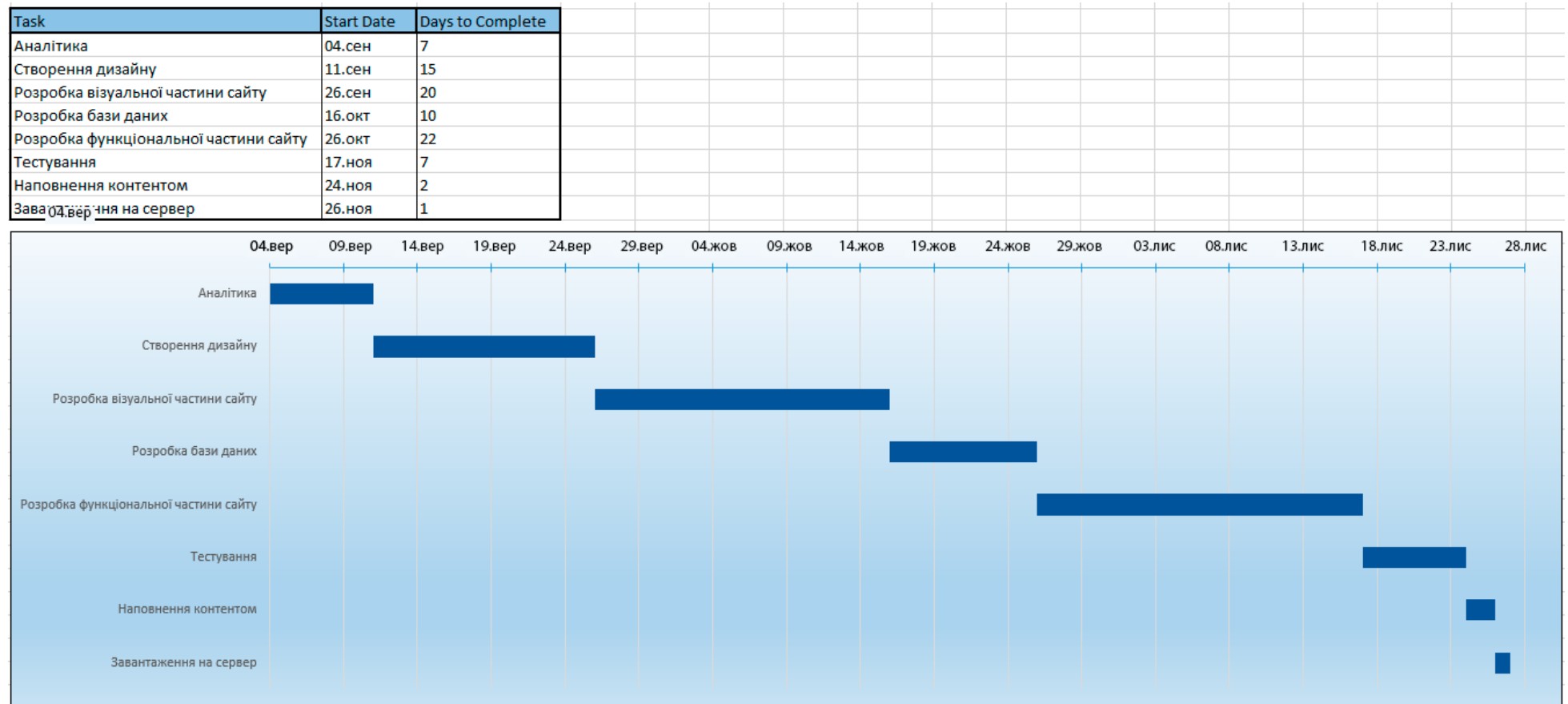


Рисунок А.3 – Діаграма Ганта

## А.4 ПЛАНУВАННЯ РИЗИКІВ ПРОЕКТУ

Ризик – це ймовірнісна подія, яка у випадку своєї появи негативно або позитивно впливає на проект.

Управління ризиком – це процес реагування на події та зміни ризиків у процесі виконання проекту. При цьому важливим є проведення моніторингу ризиків.

Процес управління ризиками включає в себе такі пункти:

- Ідентифікація ризиків (виявлення ризиків);
- Оцінювання ризиків (оцінка ймовірності та впливу);
- Заходи реагування на ризики;
- Моніторинг ризиків.

Ідентифікація ризиків – це виявлення ризиків, здатних вплинути на проект, і документальне оформлення їх характеристик. Це ітеративний процес, який періодично повторюється на всьому протязі проекту, оскільки в рамках його життєвого циклу можуть виявлятися нові ризики. Найбільш розповсюдженою характеристикою ризику є загроза або небезпека виникнення невдач у тій чи іншій діяльності, небезпека виникнення несприятливих наслідків, змін зовнішнього середовища, які можуть викликати втрати ресурсів, збитки, а також небезпеку, від якої слід застрахуватися.

Планування реагування на ризики – це процес розробки шляхів і визначення дій із збільшення можливостей і зниження погроз для цілей проекту. Даний процес зачинається після проведення якісного і кількісного аналізу ризиків. В процесі аналізу для визначення числових значень ймовірності виникнення ступеня впливу, зазвичай застосовується метод експертних оцінок. На їх основі визначається ранг ризику, як потенційний вплив ризику на проект, який оцінюється як добуток ймовірності виникнення та ступеню впливу.

Ймовірність виникнення:

1. Слабоймовірно;
2. Малоймовірно;
3. Ймовірно;
4. Вельми ймовірно;
5. Майже можливо.

Величина втрат:

1. Мінімальна;
2. Низька;
3. Середня;
4. Висока;
5. Максимальна.

Таблиця А.4 – Ймовірність втрат

Ідентифікатор ризику	Ризики	Вірогідність	Вплив
1	Ризик кібератак	2	2
2	Пошкодження даних	1	3
3	Старіння сайту під час розробки	2	2
4	Вихід з ладу обладнання	2	3

Таблиця А.5 – Ймовірність втрат

Вірогідність Вплив	1	2	3	4	5
1					
2		1, 3			
3	2	4			
4					
5					

Класифікація за ступенем впливу:

- ігноровані ( $1 \leq R \leq 4$ );
- незначні ( $5 \leq R \leq 8$ );
- помірні ( $9 \leq R \leq 11$ );
- вагомі ( $12 \leq R \leq 19$ );
- критичні ( $20 \leq R \leq 25$ ).

Класифікація за рівнем ризику:

- прийнятні ризики;
- виправданні ризики;
- недопустимі ризики.

Таблиця А.6 – Класифікація за ступенем впливу та за рівнем ризику

Ризик		Ступінь впливу	Рівень ризику
Ризик кібератак	R1	4	Ігноровані ризики
Пошкодження даних	R2	4	Ігноровані ризики
Серверні проблеми	R3	3	Ігноровані ризики
Вихід з ладу обладнання	R4	6	Незначні ризики

План по усуненню ризиків:

- Підвищення безпеки сайту та інформації користувачів.
- Резервне зберігання даних у хмарі.
- Моніторинг актуальності сайту та проведення зусиль для оновлення.
- Мати можливість продовжити роботу над розробкою сайту на іншому обладнанні.

## **ДОДАТОК Б**

**РЕАЛІЗАЦІЯ ТАБЛИЦЬ БАЗИ ДАНИХ  
для розробки кваліфікаційної роботи магістра  
«Web-орієнтована система консультування ІТ фахівців з професійних  
питань»**

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🗝️	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>questions_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>users_id</b>	bigint(20)		UNSIGNED	Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	4 <b>answers_id</b>	bigint(20)		UNSIGNED	Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	5 <b>description</b>	text	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	6 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	7 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё

Рисунок Б.1 – Таблица «answer»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🗝️	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>title</b>	varchar(255)	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>preview</b>	varchar(255)	utf8mb4_unicode_ci		Нет	/img/no-photo-available.png			Ещё
<input type="checkbox"/>	4 <b>description</b>	longtext	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	5 <b>users_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	6 <b>categories_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	7 <b>is_active</b>	tinyint(1)			Нет	0			Ещё
<input type="checkbox"/>	8 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	9 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё

Рисунок Б.2 – Таблица «article»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🗝️	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>articles_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>users_id</b>	bigint(20)		UNSIGNED	Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	4 <b>description</b>	text	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	5 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	6 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё

Рисунок Б.3 – Таблица «article\_comments»



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>title</b>	varchar(255)	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	4 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё

Рисунок Б.4 – Таблица «categories»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>themes_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>users_id</b>	bigint(20)		UNSIGNED	Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	4 <b>comments_id</b>	bigint(20)		UNSIGNED	Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	5 <b>description</b>	text	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	6 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	7 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё

Рисунок Б.5 – Таблица «comments»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>users_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>comments_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	4 <b>like</b>	tinyint(1)			Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	5 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	6 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё

Рисунок Б.6 – Таблица «comments\_likes»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>name</b>	varchar(255)	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё

Рисунок Б.7 – Таблица «countries»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>users_id</b>	bigint(20)		UNSIGNED	Нет	Нет			Ещё
<input type="checkbox"/>	3 <b>answers_id</b>	bigint(20)		UNSIGNED	Нет	Нет			Ещё
<input type="checkbox"/>	4 <b>like</b>	tinyint(1)			Нет	Нет			Ещё
<input type="checkbox"/>	5 <b>created_at</b>	timestamp			Да	NULL			Ещё
<input type="checkbox"/>	6 <b>updated_at</b>	timestamp			Да	NULL			Ещё

Рисунок Б.8 – Таблица «likes»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>title</b>	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё
<input type="checkbox"/>	3 <b>description</b>	longtext	utf8mb4_unicode_ci		Нет	Нет			Ещё
<input type="checkbox"/>	4 <b>categories_id</b>	bigint(20)		UNSIGNED	Нет	Нет			Ещё
<input type="checkbox"/>	5 <b>users_id</b>	bigint(20)		UNSIGNED	Нет	Нет			Ещё
<input type="checkbox"/>	6 <b>created_at</b>	timestamp			Да	NULL			Ещё
<input type="checkbox"/>	7 <b>updated_at</b>	timestamp			Да	NULL			Ещё

Рисунок Б.9 – Таблица «questions»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>title</b>	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё
<input type="checkbox"/>	3 <b>categories_id</b>	bigint(20)		UNSIGNED	Нет	Нет			Ещё
<input type="checkbox"/>	4 <b>questions_id</b>	bigint(20)		UNSIGNED	Нет	Нет			Ещё
<input type="checkbox"/>	5 <b>created_at</b>	timestamp			Да	NULL			Ещё
<input type="checkbox"/>	6 <b>updated_at</b>	timestamp			Да	NULL			Ещё

Рисунок Б.10 – Таблица «tags»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>title</b>	varchar(255)	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>description</b>	longtext	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	4 <b>users_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	5 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	6 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё

Рисунок Б.11 – Таблица «themes»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>name</b>	varchar(255)	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>email</b> 🔑	varchar(255)	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	4 <b>countries_id</b>	bigint(20)		UNSIGNED	Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	5 <b>photo</b>	varchar(255)	utf8mb4_unicode_ci		Нет	<i>/img/avatar.jpg</i>			Ещё
<input type="checkbox"/>	6 <b>password</b>	varchar(255)	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	7 <b>facebook</b>	varchar(255)	utf8mb4_unicode_ci		Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	8 <b>github</b>	varchar(255)	utf8mb4_unicode_ci		Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	9 <b>x</b>	varchar(255)	utf8mb4_unicode_ci		Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	10 <b>description</b>	text	utf8mb4_unicode_ci		Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	11 <b>is_admin</b>	tinyint(1)			Нет	<i>0</i>			Ещё
<input type="checkbox"/>	12 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	13 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	14 <b>google_id</b>	varchar(255)	utf8mb4_unicode_ci		Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	15 <b>github_id</b>	varchar(255)	utf8mb4_unicode_ci		Да	<i>NULL</i>			Ещё

Рисунок Б.12 – Таблица «users»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>questions_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>users_id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	4 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	5 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё

Рисунок Б.13 – Таблица «views»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 <b>id</b>	bigint(20)		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 <b>title</b>	varchar(255)	utf8mb4_unicode_ci		Нет	<i>Нет</i>			Ещё
<input type="checkbox"/>	3 <b>created_at</b>	timestamp			Да	<i>NULL</i>			Ещё
<input type="checkbox"/>	4 <b>updated_at</b>	timestamp			Да	<i>NULL</i>			Ещё

Рисунок Б.14 – Таблица «words»

## **ДОДАТОК В**

**КОД РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ  
для розробки кваліфікаційної роботи магістра  
«Web-орієнтована система консультування ІТ фахівців з професійних  
питань»**

## QuestionsController.php

Контролер для роботи з питаннями.

```

<?php

namespace App\Http\Controllers;

use App\Models\Questions;
use App\Models\Categories;
use App\Models\Tags;
use App\Models\Views;
use Illuminate\Http\Request;
use Inertia\Inertia;
use Illuminate\Support\Facades\Auth;

use App\Http\Resources\QuestionShowResource;
use App\Http\Resources\QuestionsResource;

class QuestionsController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index(Request $request)
    {
        $data = Questions::query();
        if ($request->tag) {
            $data->whereHas('tags', function ($q) use ($request) {
                $q->where('title', $request->tag);
            });
        }
        if ($request->title) {
            $data = $data->where('title', '%' . $request->title . '%');
        }
        if($request->filter) {
            switch ($request->filter) {
                case 'lasts':
                    $data = $data->orderBy('created_at', 'desc');
                    break;
                case 'noanswer':
                    $data = $data->has('answers', 0);
                    break;
                case 'popular':
                    $data = $data->withCount('views')->orderBy('views_count', 'desc');
                    break;
            }
        }
        $data = $data->get();
        $tags = Tags::limit('30')->get();
        return Inertia::render('Questions', [
            'questions' => QuestionsResource::collection($data),
            'tags' => $tags
        ]);
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()

```

```

{
  $categories = Categories::select('id as value', 'title as name')->get();
  return Inertia::render('QuestionCreate', [
    'categories' => $categories
  ]);
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
  $data = $request->all();
  $question = Questions::create([
    'title' => $data['title'],
    'categories_id' => $data['categories_id'],
    'description' => $data['description'],
    'users_id' => Auth::id()
  ]);
  foreach ($data['tags'] as $value) {
    Tags::create([
      'title' => $value,
      'categories_id' => $data['categories_id'],
      'questions_id' => $question['id'],
    ]);
  }
}

/**
 * Display the specified resource.
 */
public function show($id)
{
  if (!Views::where('users_id', Auth::id())->where('questions_id', $id)->exists()) {
    Views::create([
      'users_id' => Auth::id(),
      'questions_id' => $id
    ]);
  }
  $data = Questions::with('category', 'tags')->find($id);
  $recommend = Questions::limit('5')->get();
  $tags = Tags::where('categories_id', $data['categories_id'])->limit('30')->get();
  return Inertia::render('Question', [
    'question' => new QuestionShowResource($data),
    'recommend' => QuestionsResource::collection($recommend),
    'tags' => $tags
  ]);
}

/**
 * Show the form for editing the specified resource.
 */
public function edit($id)
{
  $data = Questions::with('tags')->find($id);
  $categories = Categories::select('id as value', 'title as name')->get();
  return Inertia::render('QuestionCreate', [
    'question' => new QuestionShowResource($data),
    'categories' => $categories
  ]);
}

```

```

    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, $id)
    {
        $data = $request->all();
        Questions::with('category')->find($id)->update([
            'title' => $data['title'],
            'categories_id' => $data['categories_id'],
            'description' => $data['description']
        ]);
        Tags::where('questions_id', $id)->delete();
        foreach ($data['tags'] as $value) {
            Tags::create([
                'title' => $value,
                'categories_id' => $data['categories_id'],
                'questions_id' => $id,
            ]);
        }
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy($id)
    {
        Questions::find($id)->delete();
    }

    function recommends()
    {
        $recommend = Questions::limit('10')->get();
        return response()->json(QuestionsResource::collection($recommend));
    }
}

```

## GitHubController.php

### Контролер авторизації за допомогою Github.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Laravel\Socialite\Facades\Socialite;
use Exception;
use App\Models\User;
use Illuminate\Support\Facades\Auth;

class GitHubController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function redirectToGithub()
    {

```



```

    return Socialite::driver('github')->redirect();
}

/**
 * Create a new controller instance.
 *
 * @return void
 */
public function handleGithubCallback()
{
    try {

        $user = Socialite::driver('github')->user();

        $finduser = User::where('github_id', $user->id)->first();

        if ($finduser) {

            Auth::login($finduser);

            return redirect()->intended('profile');
        } else {
            $newUser = User::create([
                'name' => $user->name,
                'email' => $user->email,
                'github_id' => $user->id,
                'password' => encrypt('123456dummy')
            ]);

            Auth::login($newUser);

            return redirect()->intended('profile');
        }
    } catch (Exception $e) {
        dd($e->getMessage());
    }
}
}

```

## GoogleController.php

### Контролер авторизації за допомогою Google.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Laravel\Socialite\Facades\Socialite;
use Exception;
use App\Models\User;
use Illuminate\Support\Facades\Auth;

class GoogleController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function redirectToGoogle()

```

```

{
    return Socialite::driver('google')->redirect();
}

/**
 * Create a new controller instance.
 *
 * @return void
 */
public function handleGoogleCallback()
{
    try {

        $user = Socialite::driver('google')->user();

        $finduser = User::where('google_id', $user->id)->first();

        if ($finduser) {

            Auth::login($finduser);

            return redirect()->intended('profile');
        } else {
            $newUser = User::create([
                'name' => $user->name,
                'email' => $user->email,
                'google_id' => $user->id,
                'password' => encrypt('123456dummy')
            ]);

            Auth::login($newUser);

            return redirect()->intended('profile');
        }
    } catch (Exception $e) {
        dd($e->getMessage());
    }
}
}

```

## AnswersController.php

### Контролер для відповідей.

```

<?php

namespace App\Http\Controllers;

use App\Models\Answers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AnswersController extends Controller
{
    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $data = $request->all();
        $answer = Answers::create([

```

```

        'description' => $data['description'],
        'questions_id' => $data['questions_id'],
        'answers_id' => $data['answers_id'],
        'users_id' => Auth::id()
    ]);
    return response()->json(Answers::with('user', 'answers')->find($answer['id']));
}

public function storeAI(Request $request)
{
    $data = $request->all();
    $response = Answers::create([
        'description' => $data['description'],
        'questions_id' => $data['questions_id']
    ]);
    $answer = Answers::with('answers')->find($response['id']);
    return response()->json($answer);
}

/**
 * Remove the specified resource from storage.
 */
public function destroy($id)
{
    Answers::find($id)->delete();
}
}

```

## Question.php

### КОМПОНЕНТ ЗАПИТАННЯ.

```

<script setup>
import { Head, Link } from '@inertiajs/vue3';
import { FwbButton, FwbTextarea, FwbSpinner } from 'flowbite-vue'
import Header from "../Components/Header.vue";
import Footer from "../Components/Footer.vue";
import QuestionItem from "../Components/QuestionItem.vue";
import Question from "../Components/Question.vue";
import AnswerItem from "../Components/AnswerItem.vue";
import QuestionsMin from "../Components/QuestionsMin.vue";
defineProps({
    question: {
        type: Object,
    },
    recommend: {
        type: Array,
    },
    tags: {
        type: Array,
    },
});
</script>

<template>
<Head title="Запитання" />

<div class="min-h-screen bg-bgWhite dark:bg-bgDark text-textDark dark:text-textWhite">
    <Header />
    <div class="container main">
        <div>

```

```

<Link href="/questions/create">
<fwb-button gradient="blue" class="block w-full mb-5" size="lg" shadow>Задати питання</fwb-button>
</Link>
<div class="flex flex-wrap">
  <Link :href="/questions?tag=${item.title}" v-for="item in tags" :key="item.id">
    <span
      class="bg-blue-100 text-blue-800 text-xs font-medium mr-2 mb-2 px-2.5 py-0.5 rounded dark:bg-blue-
900 dark:text-blue-300">
      #{{ item.title }}
    </span>
  </Link>
</div>
</div>
<div>
  <div class="mb-3">
    <div class="user">
      
      <a :href="/users/${question.data.user.id}">{{ question.data.user.name }}</a>
    </div>
    <Question :data="question.data" />
    <div class="flex flex-wrap mt-2">
      <Link :href="/questions?tag=${item.title}" v-for="item in question.data.tags" :key="item.id">
        <span
          class="bg-blue-100 text-blue-800 text-xs font-medium mr-2 mb-2 px-2.5 py-0.5 rounded dark:bg-blue-
900 dark:text-blue-300">
          #{{ item.title }}
        </span>
      </Link>
    </div>
    <div class="date">
      Питання задано {{ question.data.created_at }} • {{ question.data.views }} переглядів
    </div>
    <fwb-button gradient="cyan" shadow="teal" class="mt-2" @click="askAI()" v-if="question.data.user.id ==
$page.props.auth.user.id">
      <span class="flex items-center gap-1">
        Запитати в ChatGTP
        <svg aria-hidden="true" role="status" class="inline w-4 h-4 ml-1 text-white animate-spin"
viewBox="0 0 100 101" fill="none" xmlns="http://www.w3.org/2000/svg" v-if="loading">
          <path
            d="M100 50.5908C100 78.2051 77.6142 100.591 50 100.591C22.3858 100.591 0 78.2051 0
50.5908C0 22.9766 22.3858 0.59082 50 0.59082C77.6142 0.59082 100 22.9766 100 50.5908ZM9.08144
50.5908C9.08144 73.1895 27.4013 91.5094 50 91.5094C72.5987 91.5094 90.9186 73.1895 90.9186 50.5908C90.9186
27.9921 72.5987 9.67226 50 9.67226C27.4013 9.67226 9.08144 27.9921 9.08144 50.5908Z"
            fill="#E5E7EB" />
          <path
            d="M93.9676 39.0409C96.393 38.4038 97.8624 35.9116 97.0079 33.5539C95.2932 28.8227 92.871
24.3692 89.8167 20.348C85.8452 15.1192 80.8826 10.7238 75.2124 7.41289C69.5422 4.10194 63.2754 1.94025 56.7698
1.05124C51.7666 0.367541 46.6976 0.446843 41.7345 1.27873C39.2613 1.69328 37.813 4.19778 38.4501
6.62326C39.0873 9.04874 41.5694 10.4717 44.0505 10.1071C47.8511 9.54855 51.7191 9.52689 55.5402
10.0491C60.8642 10.7766 65.9928 12.5457 70.6331 15.2552C75.2735 17.9648 79.3347 21.5619 82.5849
25.841C84.9175 28.9121 86.7997 32.2913 88.1811 35.8758C89.083 38.2158 91.5421 39.6781 93.9676 39.0409Z"
            fill="currentColor" />
          </path>
        </svg>
      </span>
    </fwb-button>
  </div>

  <div class="section-header border-b-2 border-solid border-bgDark dark:border-bgWhite" v-
if="showAnswerAI">
    ВІДПОВІДЬ CHATGPT
  </div>

```

```

</div>

<div class="grid grid-cols-12 py-5" v-if="showAnswerAI">
  <div class="col-span-1">
    
  </div>
  <div class="col-span-11">
    {{ answerAI.description }}
    <div class="flex gap-2 mt-2">
      <fwb-button gradient="green-blue" @click="sendAnswerAI()" square>Опубліковати</fwb-button>
      <fwb-button color="" @click="rejectAnswerAI()">Відхилити</fwb-button>
    </div>
  </div>
</div>

<div class="section-header border-b-2 border-solid border-bgDark dark:border-bgWhite">
  ВАША ВІДПОВІДЬ НА ПИТАННЯ
</div>

<div class="grid grid-cols-12 py-5" v-if="$page.props.auth.user">
  <div class="col-span-1">
    
  </div>
  <div class="col-span-11">
    <fwb-textarea :rows="4" label="" v-model="answer.description" placeholder="Напишіть відповідь..."
      class="block f-full" />
    <div class="flex gap-2 mt-2">
      <fwb-button gradient="green-blue" @click="send()" square>Опубліковати</fwb-button>
      <!-- <fwb-button color="dark" square>Перегляд</fwb-button> -->
    </div>
  </div>
</div>

<div class="py-5" v-else>
  <div class="text-center pt-5">
    
    <div class="my-6">
      Войдите, чтобы написать ответ
    </div>
    <fwb-button :loading="loading" color="default" loading-position="suffix" outline
      @click="loading = !loading">
      Авторизуватись
    <template #suffix>
      <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20"
        xmlns="http://www.w3.org/2000/svg">
        <path clip-rule="evenodd"
          d="M10.293 3.293a1 1 0 011.414 0l6 6a1 1 0 01-1.414 1.414L14.586
11H3a1 1 0 110-2h11.586l-4.293-4.293a1 1 0 010-1.414z"
          fill-rule="evenodd" />
        </svg>
      </template>
    </fwb-button>
  </div>
</div>

<div class="section-header border-b-2 border-solid border-bgDark dark:border-bgWhite">
  ВІДПОВІДІ НА ЗАПИТАННЯ ({{ question.data.answers.length }})
</div>

<div class="text-center mt-5 text-xl" v-if="question.data.answers.length == 0">Відповідей поки немає :(

```

```

</div>

<AnswerItem v-for="answer in question.data.answers" :key="answer.id" :answer="answer" />

<div class="section-header border-b-2 border-solid border-bgDark dark:border-bgWhite">
  СХОЖИ ВІДПОВІДІ
</div>
<QuestionItem v-for="item in recommend.data" :item="item" :key="item.id" />

</div>
<div>
  <div class="title-min">Саме цікаве</div>
  <QuestionsMin />
</div>
</div>
<Footer />
</div>
</template>

<script>
export default {
  data() {
    return {
      showAnswerAI: false,
      loading: false,
      answer: {
        description: "",
        questions_id: this.question.data.id,
        answers_id: null
      },
      answerAI: {
        description: "",
        questions_id: this.question.data.id,
      }
    }
  },
  methods: {
    askAI() {
      this.loading = true;
      axios.post('/chat', {
        question: this.question.data.description
      })
        .then((response) => {
          this.loading = false;
          this.showAnswerAI = true;
          this.answerAI.description = response.data.content;
          notify({
            title: "Успіх",
            text: "Відповідь надіслано 👍",
            type: "info",
            group: "foo",
          }, 4000)
        })
        .catch(() => {
          this.loading = false;
          notify({
            title: "Помилка",
            text: "Щось пішло не так 😞",
            type: "error",
            group: "foo",
          })
        })
    }
  }
}

```

```

    }, 4000)
  })
},
sendAnswerAI() {
  axios.post('/answers-ai', this.answerAI)
    .then((response) => {
      this.showAnswerAI = false;
      this.question.data.answers.unshift(response.data);
      this.answerAI.description = "";
      notify({
        title: "Успіх",
        text: "Відповідь надіслано 👍",
        type: "info",
        group: "foo",
      }, 4000)
    })
    .catch(() => {
      notify({
        title: "Помилка",
        text: "Щось пішло не так 😞",
        type: "error",
        group: "foo",
      }, 4000)
    })
},
rejectAnswerAI() {
  this.showAnswerAI = false;
  this.answerAI.description = "";
},
send() {
  axios.post('/answers', this.answer)
    .then((response) => {
      this.question.data.answers.unshift(response.data);
      this.answer.description = "";
      notify({
        title: "Успіх",
        text: "Відповідь надіслано 👍",
        type: "info",
        group: "foo",
      }, 4000)
    })
    .catch(() => {
      notify({
        title: "Помилка",
        text: "Щось пішло не так 😞",
        type: "error",
        group: "foo",
      }, 4000)
    })
  }
}
}
</script>

```