

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних  
інформаційних систем

\_\_\_\_\_ Євген СІДЕНКО

« \_\_\_\_ » \_\_\_\_\_ 2026 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**  
**Вебсайт юридичної та психологічної допомоги для ветеранів**  
**Російсько-Української війни 2014 – 2025 рр.**

Спеціальність 122 Комп'ютерні науки  
Освітня програма «Комп'ютерні науки»

*Здобувач*

\_\_\_\_\_ Богдан ПАРХОМОВ

« \_\_\_\_ » \_\_\_\_\_ 2026 р.

*Керівник* д-р. техн. наук, професор

\_\_\_\_\_ Ірина КАЛІНІНА

« \_\_\_\_ » \_\_\_\_\_ 2026 р.

**Миколаїв – 2026**

# Чорноморський національний університет імені Петра Могили

(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	122 Комп'ютерні науки
Освітня програма	Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних  
інформаційних систем

\_\_\_\_\_ Євген СІДЕНКО  
«\_\_» \_\_\_\_\_ 2025 р.

## ЗАВДАННЯ

на кваліфікаційну роботу здобувача

**Пархомов Богдан Едуардович**

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Вебсайт юридичної та психологічної допомоги для ветеранів Російсько-Української війни 2014 – 2025 рр.

Керівник роботи: Калініна Ірина Олександрівна, доктор технічних наук, професор, професор кафедри ІС.

*(прізвище, ім'я, по батькові, посада, науковий ступінь, вчене звання)*

Затверджена наказом ЧНУ ім. Петра Могили від «25» грудня 2025 р. № 353.

2. Строк представлення кваліфікаційної роботи «\_\_» \_\_\_\_\_ 2026 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: наукові публікації, документація Angular, Node.js, REST API, технології штучного інтелекту та чат-ботів, методи NLP, засоби забезпечення безпеки вебсистем, сучасні технології веброзробки та клієнт-серверної взаємодії.

4. Перелік питань, що підлягають розробці: Аналіз предметної області та існуючих інформаційних вебсайтів підтримки ветеранів, аналіз сучасних вебтехнологій та методів побудови клієнт-серверних вебсайтів, дослідження методів використання штучного інтелекту та чат-ботів, проектування архітектури вебсайту, розробка структури бази даних та REST API, реалізація клієнтської та серверної частини вебсайту, реалізація AI-модуля для консультації та підтримки користувачів, тестування функціоналу вебсайту, аналіз результатів роботи вебсайту.

5. Перелік графічних матеріалів: Архітектура вебсайту Структура клієнтської частини, взаємодія клієнтської та серверної частини, процес обробки запитів, процес автентифікації користувача.

**Керівник роботи**

\_\_\_\_\_

*(Особистий підпис)*

Ірина КАЛІНІНА

*(Власне ім'я ПРІЗВИЩЕ)*

**Здобувач**

\_\_\_\_\_

*(Особистий підпис)*

Богдан ПАРХОМОВ

*(Власне ім'я ПРІЗВИЩЕ)*

Дата видачі завдання «22» грудня 2025 р.

## КАЛЕНДАРНИЙ ПЛАН кваліфікаційної роботи

Тема: Вебсайт юридичної та психологічної допомоги для ветеранів Російсько-Української війни 2014 – 2025рр.

№	Найменування роботи	Початок	Закінченн я	Примітки
1	Отримання завдання на виконання КР	21.12.2025	24.12.2025	Виконано
2	Аналіз предметної області та постановка задачі	25.12.2025	30.01.2026	Виконано
3	Огляд літературних джерел за темою кваліфікаційної роботи	31.01.2026	01.03.2026	Виконано
4	Огляд існуючих альтернатив до розроблюваної системи	02.03.2026	01.04.2026	Виконано
5	Реалізація обраних технологій з аналізом отриманих результатів	02.04.2026	24.05.2026	Виконано
6	Перший попередній захист КР на засіданні комісії кафедри	25.05.2026	25.05.2026	Виконано
7	Корегування роботи за результатами попереднього захисту	26.05.2026	04.06.2026	Виконано
8	Другий попередній захист КР на засіданні комісії кафедри	05.06.2026	05.06.2026	Виконано
9	Доробка та остаточне оформлення КР	06.06.2026	14.06.2026	Виконано
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	15.06.2026	19.06.2026	Виконано

**Керівник роботи**

\_\_\_\_\_  
(Особистий підпис)

Ірина КАЛІНІНА

(Власне ім'я ПРІЗВИЩЕ)

**Здобувач**

\_\_\_\_\_  
(Особистий підпис)

Богдан ПАРХОМОВ

(Власне ім'я ПРІЗВИЩЕ)

Дата складання календарного плану  
«29» січня 2026 р.

## АНОТАЦІЯ

до кваліфікаційної роботи

здобувача групи 401з ЧНУ ім. Петра Могили

Пархомова Богдана Едуардовича

на тему: **“ВЕБСАЙТ ЮРИДИЧНОЇ ТА ПСИХОЛОГІЧНОЇ ДОПОМОГИ  
ДЛЯ ВЕТЕРАНІВ РОСІЙСЬКО-УКРАЇНСЬКОЇ ВІЙНИ 2014 – 2025РР.”**

Кваліфікаційна робота присвячена розробці веб-сайту юридичної та психологічної допомоги з використанням штучного інтелекту для ветеранів Російсько-Української війни 2014-2025. Актуальність теми пов'язана з тим, що сьогодні багато ветеранів потребують швидкого доступу до психологічної та юридичної допомоги, а сучасні веб-технології зробити такі сервіси більш доступними а зручними для користувачів.

**Об'єктом** дослідження є процес надання юридичної та психологічної допомоги ветеранам засобами веб-технологій.

**Предметом** дослідження є методи та програмні засоби створення веб-сайту підтримки користувачів із використанням технологій штучного інтелекту.

**Метою** роботи є розробка веб-сайту, який забезпечить можливість отримання юридичної та психологічної допомоги підтримки ветеранам за допомогою сучасних вебтехнологій та інтелектуального чат-бота.

У роботі проведено аналіз предметної області та існуючих інформаційних систем підтримки користувачів. Розглянуто сучасні підходи до розробки веб-сайту, клієнт-серверну архітектуру, REST API, а також забезпечення безпеки веб-сайтів. Окрему увагу приділено використанню технологій штучного інтелекту та чат-ботів для автоматизації взаємодії з користувачем.

Практичне значення роботи полягає у можливості використання розробленого веб-сайту для спрощення доступу ветеранів до консультаційної допомоги та інформаційної підтримки.

Кваліфікаційна робота містить 83 сторінок, 16 рисунків, 24 таблиць, 30 джерел посилання, 1 додаток.

**Ключові слова:** веб-сайт, інформаційна система, штучний інтелект, чат-бот, REST API, Angular, Node.js, психологічна підтримка, еридика допомога.

## ABSTRACT

to the qualification work by the student of the group 401z of Petro Mohyla Black Sea  
National University

Parkhomov Bohdan

Website for legal and psychological assistance for veterans of the Russian-Ukrainian  
war 2014-2025

The qualification work is devoted to the development of a website for legal and psychological assistance of veterans using artificial intelligence technologies. The relevance of the topic is related to the fact that today many veterans need quick access to psychological and legal assistance, while modern web technologies make such services more accessible and convenient for users.

**The object** of the research is the process of providing legal and psychological assistance to veterans by means of web technologies.

**The subject** of the research is methods and software tools for developing a web-based support system using artificial technologies/

**The purpose** of the work is to develop a website that will provide veterans with the opportunity to receive legal and psychological assistance using modern web technologies and an intelligent chatbot.

The paper analyzes the subject area and existing user support information systems. Modern approaches to website development, client-server architecture, REST API, and methods of ensuring website security are considered. Particular attention is paid to the use of artificial intelligence technologies and chatbots for automating interaction with users/

The practical significance of the work lies in the possibility of using the developed system to simplify veterans access to consulting assistance and informational support.

The qualification work contains 83 pages, 16 figures, 24 tables, 30 references, 1 appendice.

**Keywords:** website, information system, artificial intelligence, chatbot, REST API, Angular, Node.js, psychological support, legal assist

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	4
ВСТУП .....	5
1 АНАЛІЗ ВЕБ-САЙТІВ ДЛЯ ПІДТРИМКИ ВЕТЕРАНІВ ТА ПОСТАНОВКА ЗАДАЧІ.....	6
1.1 Стан проблеми підтримки ветеранів в Україні .....	6
1.2 Аналіз існуючих інформаційних ресурсів та веб-сайтів з підтримки ветеранів 8	
1.3 Недоліки існуючих веб-сайтів з підтримки ветеранів .....	10
1.4 Постановка задачі .....	12
Висновки до розділу 1 .....	15
2 ОБГРУНТУВАННЯ ТЕХНОЛОГІЧНИХ РІШЕНЬ ТА МЕТОДІВ РОЗРОБКИ ВЕБ-САЙТУ .....	16
2.1 Архітектура веб-сайту допомоги ветеранам .....	16
2.2 Вибір та обґрунтування технологій розробки веб-сайту .....	19
2.3 Використання технологій штучного інтелекту у веб-сайті підтримки ветеранів 24	
2.4 Забезпечення безпеки та надійності веб-сайту .....	27
Висновки до розділу 2 .....	31
3 ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБ_САЙТУ ЮРИДИЧНОЇ ТА ПСИХОЛОГІЧНОЇ ДОПОМОГИ З ВИКОРИСТАННЯМ ШІ ДЛЯ ВЕТЕРАНІВ	32
3.1 Проектування структури веб-сайту .....	32
3.2 Проектування функціональної структури веб-сайту .....	35
3.3 Проектування структури бази даних веб-сайту .....	39

3.4 Розробка клієнтської частини та інтерфейсу веб-сайту.....	43
3.5 Розробка серверної частини та бази даних.....	49
3.6 Реалізація модуля штучного інтелекту .....	52
3.7 Аналіз результатів розробки .....	55
Висновки до розділу 3 .....	57
4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ВЕБ-САЙТУ .....	59
4.1 Методики та функціональне тестування веб-сайту.....	59
4.2 Аналіз ефективності та оцінка роботи веб-сайту .....	61
4.3 Інструкція користувача.....	63
Висновок до розділу 4 .....	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	69
ДОДАТОК А.....	72
Лістинг програмної реалізації основних компонентів вебсайту .....	72

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AI	- Artificial Intelligence, штучний інтелект
API	- Application Programming Interface, Інтерфейс програмування
CSRF	- Cross-Site Request Forgery, підробка міжсайтових запитів
HTTP	- HyperText Transfer Protocol, протокол передачі гіпертексту
HTTPS	- HyperText Transfer Protocol Secure, захищений протокол передачі гіпертексту
JSON	- JavaScript Object Notation, формат обміну даних
JWT	- JSON Web Token, токен авторизації та автентифікації
NLP	- Natural Language Processing, обробка природної мови
REST	- Representation State Transfer, архітектурний стиль взаємодії клієнта та сервера
SPA	- Single Page Application, односторінковий веб-сайт
SQL	- Structured Query Language, мова структурованих запитів до баз даних
UI	- User Interface, інтерфейс користувача
UX	- User Experience, користувацький досвід взаємодії із сайтом
XSS	- Cross-Site Scripting, тпп атаки пов'язаний із виконанням шкідливо коду на стороні клієнта
DB	- Database, база даних
HTML	- HyperText Markup Language, мова розмітки веб-сайту
CSS	- Cascading Style Sheets, каскадні таблиці стилів
JS	- JavaScript, мова програмування веб-сайтів
TS	- TypeScript, мова програмування на основі JavaScript
URL	- Uniform Resource Locator, адреса веб-сайту

## ВСТУП

На теперішній час тема надання комплексної допомоги ветеранам є надзвичайно актуальною. Після початку Російсько-Української війни у 2014 році, яка у 2022 році переросла у повномасштабне вторгнення з боку Росії, та триває по цей день, значно зросла кількість військовослужбовців, які після завершення служби, або списання зі служби по тим чи іншим причинам, потребують соціальної адаптації, психологічної підтримки, та юридичного супроводу, для всіх ветеранів важливим питанням є отримання пільг, оформлення документів, отримання іншої допомоги від держави та інші юридичні питання.

Традиційні методи та способи надання допомоги не забезпечують належної, зручної та швидкої допомоги. Тому використання сучасних веб-технологій та засобів штучного інтелекту є перспективним напрямком для створення онлайн-платформи, здатної забезпечити оперативне консультування, автоматизацію обробки звернень та зменшення навантаження на профільних спеціалістів.

Метою роботи є розробка веб-сайту, який надає інтерактивну юридичну та психологічну допомогу військовослужбовцям та ветеранам на базі штучного інтелекту.

Об'єктом дослідження є процес надання юридичної та психологічної допомоги засобами веб-технологій.

Предметом дослідження є методи, інструменти та програмні рішення для створення інтелектуального веб-сайту з інтеграцією штучного інтелекту у сайт для підтримки ветеранів в юридичній та психологічній сфері.

## **1 АНАЛІЗ ВЕБ-САЙТІВ ДЛЯ ПІДТРИМКИ ВЕТЕРАНІВ ТА ПОСТАНОВКА ЗАДАЧІ**

### **1.1 Стан проблеми підтримки ветеранів в Україні**

У наш час та умовах питання підтримки [1] військовослужбовців та ветеранів Російсько-Української війни є одним із найбільших важливих соціальних напрямів розвитку в державі. Внаслідок цієї війни, що почалась в 2014 році, переросла у повномасштабне вторгнення в 2022 році, дуже велика кількість громадян які брали участь у бойових діях та після завершення служби повертаються до цивільного життя, але поняття завершення служби, є обширним у цих умовах, хтось звільнився по стану здоров'я, хтось із за поранення, хтось після полону, тому верста ветеранів які потребують всебічної та нагальної підтримки дуже велика. Саме тому проблема комплексної допомоги ветеранам набуває особливої актуальності.

Після завершення військової служби ветерани стикаються зі складнощами соціальної адаптації. Перехід від військового середовища та постійної зони бойових дій, що вже увійшли у життя як звичка, до цивільного способу життя супроводжується зміною звичного режиму, необхідністю пошуку нового місця роботи, відновлення сімейних та соціальних зв'язків та також вирішення купи організаційних питань. Ці всі чинники створюють додаткове психологічне навантаження і так після складних психологічних та моральних навантажень на службі.

Однією з проблем є необхідність отримання своєчасної юридичної допомоги. Ветерани часто звертаються з питаннями щодо оформлення статусу учасника бойових дій, отримання державних пільг, соціальних виплат, проходження медичної реабілітації, забезпечення житлом, працевлаштування, пенсійного забезпечення та захисту трудових прав. Значна кількість нормативно-правових документів, змін у законодавстві та бюрократичних процедур це фактори які ускладнюють самостійне вирішення таких питань.

Не менш важливим напрямом є психологічна підтримка ветеранів. Після

перебування в зоні бойових дій людина може стикатися з наслідками тривалого стресу, емоційного виснаження, тривожності, порушення сну, депресивних станів, та труднощів у спілкуванні з оточенням. У багатьох випадках виникає потреба у консультації психолога, проходження реабілітації, відновлення та участі у групах підтримки.

Окремою проблемою є доступність таких послуг. Не всі ветерани мають можливості для того щоб звертатись або відвідувати спеціалістів, центри допомог, через можливо фінансові обмеження, місце проживання, стан здоров'я чи просто відсутність достатньої кількості центрів допомоги, спеціалістів у населеному пункті. Окрім того більша частина ветеранів не володіє інформацією про ці всі програми, методи їх отримання та інші можливості, через недостатнє інформування ветеранів.

У сучасному світі дуже велику роль відіграють цифрові технології, які дозволять зробити допомогу доступнішою та швидшою. Використання веб-сайтів, мобільних застосунків, онлайн консультацій, та інших автоматизованих систем дає змогу отримати необхідну інформацію дистанційно у зручний час, зручному форматі, та без необхідності особисто відвідувати профільного спеціаліста чи установу. Особливу перспективу розвитку для цих веб-сайтів є застосування в них штучного інтелекту.

Створення єдиної інформаційної веб-платформи для ветеранів дозволить об'єднати юридичну та психологічну допомогу в одному сервісі. Такий веб-сайт може містити базу знань, онлайн-чат, можливість подання звернень, запис на консультацію, персональний кабінет користувача та інші корисні функції. Це сприятиме зменшенню часу очікування відповіді, підвищення зручності користування та ефективнішому використанню ресурсів фахівців.

Виходячи з цього, проблема підтримки ветеранів в Україні є комплексною та охоплює різні фактори такі як юридична допомога, психологічна, соціальна та організаційна. Зростання кількості ветеранів з кожним днем та необхідність зручного та швидкого доступу до якісної допомоги обумовлюють доцільність

створення сучасних веб-орієнтованих інформаційних веб-сайтів із використанням технологій штучного інтелекту. Це дозволить підвищити рівень доступності послуг і сприятиме успішній адаптації ветеранів до цивільного життя після звільнення зі служби.

## **1.2 Аналіз існуючих інформаційних ресурсів та веб-сайтів з підтримки ветеранів**

На теперішній час в Україні функціонує значна кількість державних, приватних та громадських формувань, які направлені на надання допомоги ветеранам та членам їх сімей [2]. Основними напрямками діяльності таких ресурсів є інформування щодо соціальних гарантій, юридичних консультацій, психологічної підтримки, медичної реабілітації, працевлаштування та інших аспектів соціальної адаптації після завершення військової служби. Не дивлячись на наявність великої кількості різних формувань та сервісів, проблема створення єдиного зручного веб-сайту підтримки залишається відкритою.

Головним джерелом інформації залишаються державні веб-портали та сайти профільних формувань та установ. На таких сайтах розміщується інформація щодо різних аспектів, таких як отримання статусу учасника бойових дій, оформлення пільг, державних виплат, медичних програм, соціальних гарантій, та інших важливих питань. Головна перевага цих ресурсів, це достовірність та актуальність інформації, але якщо подивитись на досвід користування цими веб-сайтами, користувачі скаржаться на складну структуру сайтів, велику кількість нормативних документів, які не всім людям зрозумілі повноцінно.

Окремо є різні громадські, приватні, волонтерські організації, які створюють власні інформаційні платформи для підтримки ветеранів. Такі сервіси часто пропонують безкоштовні юридичні консультації, психологічну допомогу, програми перекваліфікації для пошуку нової роботи та соціальну адаптацію. Перевагою цих ресурсів є гнучкість, швидко оновлення інформації, та орієнтація на реальні потреби користувачів, але можливості цих організацій зазвичай

обмежені фінансуванням або кадровими ресурсами.

Також поширеними є гарячі лінії та контакт центри, де ветерани можуть отримати консультацію по телефону чи через електронну пошту. Подібний формат є зручним для оперативного зв'язку, але він залежить від графіка роботи операторів та кількості доступних спеціалістів. У години підвищеного попиту можуть виникати затримки у наданні відповідей.

З розвитком цифрових технологій дедалі популярніше стають онлайн-консультації та чат-сервіси. Частина організацій використовує месенджери, чат-боти або спеціальні форми звернень на сайтах. Це дозволяє швидко поставити питання та отримати відповідь дистанційно, але нюанс таких сервісів що вони виконують лише базові функції та не забезпечують повного супровіду користувача.

Окремим напрямом є сервіси та формування для психологічної допомоги та супроводу. Зазвичай вони можуть включати онлайн запис до психолога, дистанційні консультації, звичайні інформаційні матеріали із загальної інформацією щодо боротьби зі стресом, визначення емоційного стану та адаптацій у соціумі. Такі ресурси є корисним, проте психологічна часто представлена окремо від юридичних сервісів, що змушує користувача звертатися до різних платформ.

Для більш наочного порівняння краще порівняти можливості та основні типи існуючих веб-сайтів та формувань (див. табл. 1.1)

Аналізуючи всю вище сказану інформацію, та ту яка наведена у Таблиці 1.1 – порівняння типів існуючих веб-сайтів, наразі існуючі ресурси вирішують завдання по окремо та не мають комплексного підходу та повного супровіду ветеранів. Одні спеціалізовані на юридичній допомозі, інші на психологічній, але мають свої недоліки, так як обмежені кадровим персоналом, фахівцями та фінансуванням, та не використовують можливості штучного інтелекту. Тому постає потреба у створенні єдиного веб-сайту на якій можна буде отримати комплексну допомогу та супровід з різних питань, яка буде доступна цілодобово та буде зручна у користуванні для користувачів.

Таблиця 1.1 – Порівняння типів існуючих веб-сайтів.

Тип ресурсу	Юридична допомога	Психологічна допомога	Онлайн доступ	Автоматизація процесів
Державні портали	Так	Частково	Так	Низькопродуктивна
Громадські організації	Так	Так	Так	Середньопродуктивна
Гарячі лінії	Так	Так	Частковий	Низькопродуктивний
Чат-боти та месенджери	Часткова	Часткова	Так	Високопродуктивна
Запропонована система комплексної допомоги	Так	Так	Так	Високопродуктивна

### 1.3 Недоліки існуючих веб-сайтів з підтримки ветеранів

Враховуючи наявність великої кількості інформаційних ресурсів про які ми говорили в минулому розділі та сервісів підтримки ветеранів, їх ефективність є дуже обмеженою через низку суттєвих недоліків. Аналіз показує, що більшість платформ не забезпечує, комплексного, зручного та швидкого доступу до необхідної допомоги.

Ключова проблема є відсутність єдиного інформаційного простору. Ветерани змушені звертатися до різних ресурсів для вирішення окремих питань чи то юридичних чи психологічних чи інших. Це ускладнює процес отримання допомоги, потребує додаткового часу та створює незручності для користувача.

Іншою проблемою є складність користування існуючими веб-ресурсами. Державні портали часто мають перевантажений інтерфейс, складну структуру та велику кількість нормативних документів, що значно ускладнює пошук необхідної інформації. Для користувачів без достатнього рівня цифрової грамотності це може

стати перешкодою для використання цих ресурсів.

Також є проблема у низькому рівні автоматизації процесів обробки звернень чи запитів. У багатьох випадках всі звернення та запити обробляються вручну і це створює дуже велику проблему швидкості обробки звернень, якості бо враховуючи людський фактор, користувача можуть направити не туди, та він буде гаяти свій час а можливо і фінанси.

З цього впливає ще одна проблема, обмежений доступ у режимі 24 години на добу. Більшість платформ працюють лише у робочі дні та час. Це створює труднощі для ветеранів які потребують негайної підтримки, особливо психологічної, крім того спостерігається відсутність персонізованого підходу до користувача. Більшість веб-сайтів не враховують індивідуальні потреби користувача, ситуацію, історію звернень. Це значно знижує ефективність надання допомоги та ускладнює процес надання допомоги.

Також важливим аспектом є розділені платформи юридичної та психологічної допомоги, внаслідок чого користувач має самостійно шукати кожен ресурс, що призводить до помилок або затримок у отриманні допомоги.

Ще як проблему варто виділити недостатнє використання сучасних технологій, зокрема штучного інтелекту. Більшість сервісів не використовують можливості автоматичного аналізу звернень, генерації відповідей або рекомендацій, що значно зменшує та обмежує їх функціонал.

Найголовнішою проблемою є недостатній рівень інформованості ветеранів, що вони просто напросто не знають про існування потрібних їм сервісів або платформ.

Аналізуючи та узагальнюючи інформацію надано вище, основні недоліки рішень можна представити у вигляді таблиці (див. табл. 1.2)

Таблиця 1.2 – Основні недоліки існуючих веб-сайтів

Недолік	Опис
Відсутність єдиної системи	Різні сервіси для різних задач
Складний інтерфейс	Незручна навігація та пошук
Низька автоматизація	Ручна обробка звернень
Обмежений режим роботи	Відсутність 24/7 підтримки
Відсутність персоналізації	Не враховуються потреби користувача
Відсутність ШІ	Немає інтелектуальних функцій
Низька обізнаність	Користувачі не знають про сервіси

Таким чином, аналіз існуючих рішень показує, що на сьогодні відсутній універсальний веб-сайт який б забезпечував комплексну, ефективну та доступну підтримку ветеранів. Це обумовлює необхідність розробки нового веб-орієнтованого рішення, яке усуну зазначені недоліки та забезпечить комплексний сервіс.

#### 1.4 Постановка задачі

Базуючись на основі проведеного аналізу у минулих пунктах, досліджені існуючих інформаційних формувань підтримки ветеранів виходить, що сучасні рішення не дають повного забезпечення потреб користувачів. Головною визначеною проблемою було виявлено, не надання комплексного супроводу та допомоги ветеранам у питаннях юридичної та психологічної допомоги, а також малу доступність цих формувань та сервісів, за рахунок низької швидкості обробки звернень без підтримки 24 години на добу.

Виходячи з цього дуже доцільно розробити новий веб-сайт підтримки та супроводу ветеранів, яка б вирішувала ці всі проблеми, а тобто мала комплексний підхід у юридичній, психологічній допомозі та супроводі а також була просто у користуванні і доступна 24 години на добу 365 днів в рік.

Метою цієї роботи є розробка веб-сайту, який надає можливість отримати

комплексну юридичну та психологічну допомогу, без потреби використань інших ресурсів та сервісів за допомогою сучасних технологій та використання штучного інтелекту як основу для сайту.

Об'єктом дослідження є процес та підходи для надання якісної та кваліфікованої юридичної та психологічної допомоги.

Предметом досліджень є різні методи, підходи, моделі, алгоритми, програмні засоби створення сайту юридичної та психологічної допомоги для ветеранів Російсько-Української війни.

Для досягнення поставленої мети у цій роботі необхідно вирішити:

- узагальнити результати аналізу предметної області та сформулювати основні вимоги до веб-сайту;
- визначити функціональні можливості майбутнього веб-сайту;
- обґрунтувати доцільність використання технологій штучного інтелекту;
- обрати архітектурний підхід до побудови веб-сайту;
- визначити технології реалізації програмного продукту;
- спроектувати структуру веб-сайту та її основні компоненти;
- реалізувати основні функціональні модулі;
- провести тестування веб-сайту та оцінити результати її роботи.

Розроблюваний сайт буде являти собою веб-платформи, яка буде забезпечувати інтеграції юридичної та психологічної допомоги в одному сервісі. Основна ідея це створення сервісу в якому користувач може отримати комплексний підхід та допомогу зі всіх питань які його цікавлять, без необхідності постійно переключатись між різними сервісами.

Веб-сайт повинен буде забезпечувати швидкий, та легкий доступ до різної інформації з різних питань щодо юридичних і так само щодо психологічних, а також мати можливість для контакту з фахівцями та автоматичної обробки запитів. Головна особливість сервісу це вбудований штучний інтелект, для отримання користувачами різної допомоги, який зможе на початкових етапах до зв'язку з

фахівцем можливо вирішити всі запитання які були у ветерана який прийшов за допомогою.

Функціональна структура веб-сайту буде складатися з декількох основних підходів, і кожен з них буде відповідати за різні типи взаємодії користувача і сервісу. Перше це реалізація можливості реєстрації та логіну користувача, це допомже сервісу зберігати дані про користувача та більш чітко налаштовувати персоналізований підхід до кожного користувача. Наступне це юридична допомога, вона дає можливість подання запитів до сайту, надання необхідної інформації за запитом, консультація від штучного інтелекту та за потреби можливість записатись до відповідного фахівця. Також важливим є психологічна допомога, вона буде так само як і юридична, при поданні запиту, користувач буде отримувати певну інформацію, консультацію від штучного інтелекту та за потреби можливість запису до фахівця. Також окрім відповідей на запити, буде окремий чат зі штучним інтелектом де можна буде поспілкуватись та отримати якісь поради від нього. Для адміністраторів та фахівців у сервісі, будуть можливості керувати обліковими записами користувачів, запитами та обробляти їх.

З технічної точки зору та підходу до реалізації найкраще використовувати клієнт-серверну архітектуру, вона надасть можливість розподілити навантаженість веб-сайту окремо на фронтенд та бекенд, та клієнтська частина відповідає за інтерфейс який буде бачити користувач та матиме можливість з ним взаємодіяти, а серверна буде обробляти різні запити, взаємодіяти з базою даних та виконувати інші процеси. Взаємодія між клієнтом та сервером може бути реалізована використовуючи підхід REST API, що дасть можливість легкої, швидкою та головне підтримуваної взаємодії. База даних буде використовуватись для зберігання інформації про користувачів, про їх запити, звернення та інші дані. Вибір конкретної бази даних буде залежати від можливостей цієї бази даних в швидкій та легкопідтримуваній роботі, та об'єму записаної в неї інформації. Для інтеграції штучного інтелекту можливо використати два підходи, це при допомозі API або доступних бібліотек, але при реалізації через бібліотеки більш складно

підтримувана веб-система, тому найвірогідніший варіант це підключення зовнішніх API.

Також зі сторони нефункціональних вимог, по перше веб-сайт має бути доступний 24 години на добу 365 днів в рік для будь кого, чи то ветеран який прийшов за допомогою, чи то фахівець який вирішив обробити запити. Також має бути повна безпека даних користувачів, це один з важливих пріоритетів, тому для веб-сайту будуть використовуватися сучасні методи входу та захисту інформації користувача. Ще один аспект це можливість легкої підтримки роботоспособності сайту, та його масштабування для більших можливостей.

Очікуваним результатом після проведення всіх робіт по дослідженню та розробці сайту, доступний сервіс з комплексним підходом до юридичної та психологічної допомоги. В моїх очікуваннях, що після розробки цього веб-сайту більше ветеранів зможуть отримувати більше потрібної їм допомоги, краще адаптуватись до цивільного життя. Сформульована раніше постановка задачі визначає які в нас є основні напрямки розробки цього сайту, а також потрібність такого веб-сайту у широких масах серед ветеранів, та саму доцільність розробки веб-сайту для ветеранів Російсько-Української війни.

## **Висновки до розділу 1**

У першому розділі ми провели аналіз сфери, що пов'язана з наданням юридичної та психологічної допомоги. Розглянули різні сучасні підходи для надання такої допомоги ветеранам та визначили основні проблеми та труднощі у цьому. Проаналізували вже існуючі системи, сайти, чат боти та головна проблема усіх, що вони не надають комплексної допомоги ветеранам у їх питаннях, тому це створює незручності користування, знаходження інформації та інше. Разом з цією проблемою є проблема низької швидкості обробки запитів, із за недостатку фінансування чи фахівців для цього. Після проведеного аналізу було сформовано план задачі для створення такого веб-сайту, який вирішує ці всі недоліки і цими самими недоліками було обгрунтовано необхідність створення такого веб-сайту.

## **2 ОБГРУНТУВАННЯ ТЕХНОЛОГІЧНИХ РІШЕНЬ ТА МЕТОДІВ РОЗРОБКИ ВЕБ-САЙТУ**

### **2.1 Архітектура веб-сайту допомоги ветеранам**

В сучасних умовах та світі сайти та інші різні веб-системи є ключовим у поширенні різної інформації для користувачів та забезпечення доступу до неї улюбий момент. Основна перевага цих сайтів, що їх можна використовувати не встановлюючи на власній пристрій, а просто при наявності підключення до інтернету, користувач може підключитись чи зайти на сайт з пристрою який у нього є це або комп'ютер, ноутбук, планшет, телефон, а враховуючи розвиток технологій це може бути і телевізор або автомобіль та його медіа система та інше. І для цього не потрібні додаткові якісь програмні забезпечення, тому вибір архітектури веб-сайту є дуже важливим при розробці конкретного продукту.

Архітектура сама по собі визначає фундамент системи, тобто якими способами взаємодіють компоненти, можливості продуктивності веб-сайту, його можливості до розвитку, масштабованості та покращення і зручності підтримки вже існуючого продукту. Для розробки сайту юридичною та психологічною допомогою ветеранам, виходячи зі всіх факторів найдоцільніше використовувати клієнт-серверну архітектуру сайту.

Відповідно до клієнт-сервісної архітектури веб-сайт розподілений на дві частини клієнтська, це та частина яка відображається користувачу, яка зустрічає його при вході на сайт та з якою він може взаємодіяти, натискаючи різні кнопки та інші елементи інтерфейсу. На серверній частині вже йде обробка більшості дій користувача, сервер взаємодіє з клієнтом беручи звітти інформацію, що йому зробити та взаємодіє з базою даних, надсилаючи дані на клієнт частину, загальну архітектуру веб-сайту відображено (див. рис. 2.1)

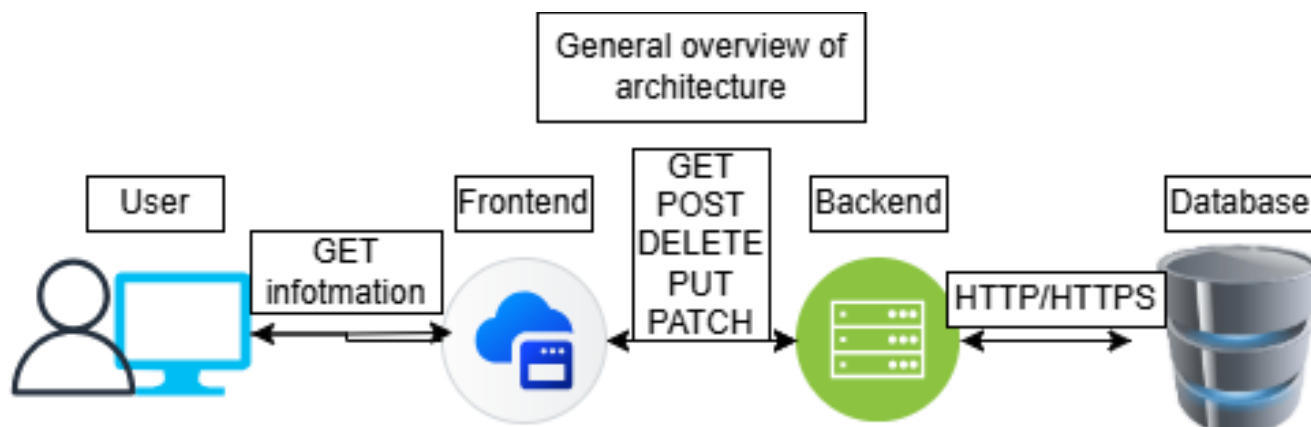


Рисунок 2.1 – Архітектура веб-сайту

Найголовніше в клієнт серверній архітектурі, одночасно це і її перевага, можливість для розробки окремих компонентів окремо в системі. Це надає таку можливість, щоб змінювати клієнтську або серверну частину, без глобальної зміни всієї архітектури.

Для розробки клієнтської частини веб-сайту, найкращий варіант це використовувати підхід Single Page Application (SPA). Головна перевага цього підходу, що по суті веб-сайт на стороні клієнта запускається лише один раз, та всі наступні переходи між сторінками на сайті, йдуть без підвантаження та перезавантаження сторінок, це в рази дає швидшу та більш зручну роботу з сайтом. SPA системи взаємодіють з сервером за допомогою API, отримуючи дані з серверу у досить розповсюдженому форматі JSON, що зручніше для обробки цих даних на клієнтській частині та показувати їх користувачу.

Для взаємодії між клієнтом та сервером краще використовувати REST архітектуру запитів до сервера. Вона зручна тим, що використовує стандартні HTTP методи, такі як GET, POST, PUT, DELETE, PATCH, це поширена архітектура запитів до сервера, і використання її значно спрощує підтримку, налаштування, та масштабування веб-сайту.

Виходячи з вище сказаного та перерахованого така архітектура як клієнт-серверна, з використанням SPA та REST API, дозволить підтримувати високу

продуктивність на веб-сайті, зручність до налаштувань та впроваджень нових рішень, і тому такий підхід для розробки сайту є достатньо обгрунтованим, та буде використаний у цій роботі. Аналіз підходів до побудови архітектури веб-сайту (див. табл. 2.1)

Таблиця 2.1 – Аналіз підходів до побудови архітектури веб-сайту

Джерело	Напрямок дослідження	Основні особливості	Практичне значення
Pro Angular [3]	SPA-архітектура	Компонентний підхід	Побудова frontend
Node.js, MongoDB and Angular [4]	Full-stack архітектура	Взаємодія frontend/backend	Побудова веб-сайтів
Node.js Performance [5]	Серверна архітектура	Асинхронна модель	Обробка HTTP-запитів
Angular Documentation [6]	Архітектура Angular	Модульність та компоненти	SPA-додатки
Node.js Documentation [7]	Серверна платформа	Робота з HTTP та API	Серверна логіка
Express.js Documentation [8]	REST API	Маршрутизація запитів	Побудова API
REST API Guide [9]	REST-архітектура	Стандартизація взаємодії	Інтеграція компонентів
W3C Web Standards [10]	Веб-стандарти	Сумісність веб-технологій	Розробка веб-сайту

## 2.2 Вибір та обґрунтування технологій розробки веб-сайту

Після того як ми визначились із загальною архітектурою веб-сайту та обґрунтували доцільність її використання, наступний крок це вибір технологій для розробки веб-сайту за клієнт-серверною архітектурою. Це також є важливим етапом, бо вибір технологій впливає на те як веб-сайт буде швидко працювати, легко підтримуватись, та як довго її технологічний стек буде залишатись актуальним. Для кожної частини цього веб-сайту були обрані відповідні технології, які будуть оптимальними та підходять під вимоги обраної архітектури такі як швидкість роботи, оптимальність підтримки, масштабування та розвиток веб-сайту.

Для того щоб реалізувати клієнтську частину веб-сайту було обрано фреймворк Angular, який як раз широко використовується для створення SPA сайтів. Angular, дає можливість як раз створити динамічний, сучасний та швидкий інтерфейс на основі SPA, те що вже було обговорено раніше чому ця архітектура є найкращою для нашого веб-сайту. Ключова особливість цього фреймворку у використанні компонентного підходу до створення інтерфейсів. За рахунок компонентів, в самій структурі Angular додатку, це все окремі незалежні частини, які відповідають кожна за свій функціонал. Це те що значно спрощує подальшу розробку, підтримку, масштабування, та тестування нашої клієнтської частини сайту. Angular також використовує мову програмування TypeScript, яка є надбудовою над JavaScript і забезпечує статичну типізацію у коді, що значно зменшує кількість помилок при роботі веб-системи, простіше тестування, та головне це забезпечує більшу безпеку веб-сайту.

Серед основних переваг Angular та його можливостей які є важливим фактором при виборі технології для клієнтської частини веб-сайту:

- двостороннє зв'язування даних, яке забезпечує автоматичне оновлення інтерфейсу без перезавантаження сторінки при зміні даних;

- вбудовану систему маршрутизації для переходу між сторінками без перезавантаження сторінок;
- модульну структуру яка дозволяє легше організувати код та вести його підтримку;
- HTTP-клієнт для взаємодії з сервером;
- підтримку форм та валідації введених даних.

Використання фреймворку Angular дозволяє зі своїми можливостями створити легко підтримувану, швидку, продуктивну клієнтську частину, що є важливим для створення такої складності веб-сайту, структура клієнтської частини (див.рис.2.2).

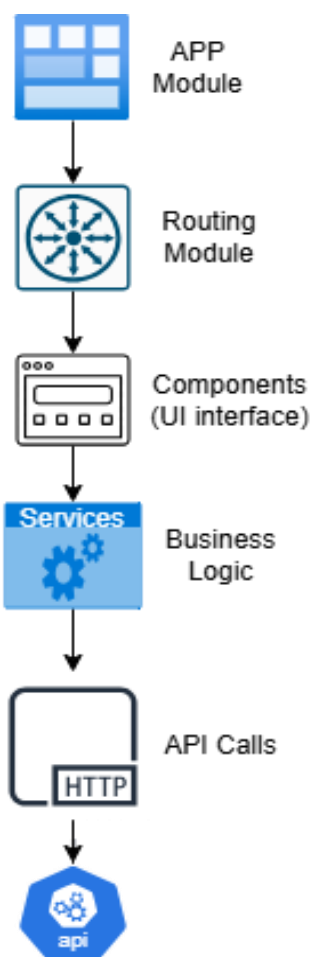


Рисунок 2.2 – Структура клієнтської частини

При виборі технології для серверної частини веб-сайту, було обрано технології

Node.js. Вона базується на JavaScript і дозволяє виконувати код на стороні сервера, також є високопродуктивною, та за рахунок внутрішніх технологій не блокуючого вводу-виводу забезпечує безперебійну та стабільну роботу сервера відразу з великою кількістю запитів, тим самим дозволивши нашому веб-сайту одночасно обробляти велику кількість запитів від користувачів, та швидко надавати відповіді на їх запити. Також додатково до Node.js доцільно використовувати фреймворк Express він спрощує роботу з REST API маршрутизацію запитів, та має встроєні розширені можливості для Node.js.

З ключових переваг для використання саме Node.js:

- асинхронна обробка запитів;
- висока продуктивність;
- використання по факту єдиної мови програмування що на клієнті, що на сервері, що забезпечую легшу підтримку всього веб-сайту у розробці та потребує меншу кількість розробників;
- велика кількість встроєних та не тільки бібліотек для розробки, що спрощують та пришвидшують розробку та підтримку веб-сайту.

Взаємодія клієнта та сервера через API (див.рис. 2.3)

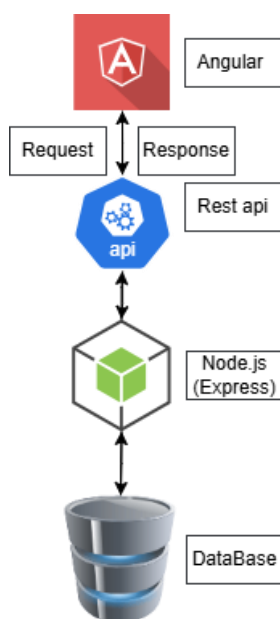


Рисунок 2.3 – Взаємодія клієнтської та серверної частини

При виборі бази даних доцільно розглянути та використовувати реляційну базу даних, наприклад як PostgreSQL або MySQL. Вибір саме реляційної бази даних потрібне для роботи зі структурованими даними, та чіткою підтримкою зв'язків між таблицями всередині бази даних. База даних у цьому випадку використовується для зберігання інформації про користувачів, звернень, повідомлень, а також історію взаємодії з системою та зберігання інших даних отримуваних від користувача.

З основних переваг реляційних баз даних це:

- забезпечення цілісності даних;
- підтримка чітких зв'язків між таблицями;
- можливість виконання складних запитів до бази даних;
- надійність та стабільність роботи;
- легкість у розгортанні та налаштуванні бази даних.

Враховуючи ці переваги реляційних баз даних було доцільно зупинити свій вибір саме на них.

Для того щоб ефективно та зручно комунікувати між серверною та клієнтською частиною використовується розповсюджена архітектура REST API запитів, що дає можливість передавати дані у форматі JSON який є широко розповсюдженим та більш стандартизованим у обміні інформації, тому для його обробки є багато встроєних бібліотек та можливостей, що значно спрощує роботу з таким форматом даних, які ми будемо отримувати від нашої серверної частини веб-сайту, які надсилаються та отримують за допомогою REST API запитів, а також такий підхід забезпечує:

- гнучкість веб-сайту;
- можливість масштабування;
- простоту інтеграції з іншими сервісами.

Тому враховуючи переваги цієї архітектури яка має назву REST API, доцільно її використовувати для забезпечення комунікації у нашому веб-сайті з клієнт-

серверною архітектурою.

Виходячи з усього вище перерахованого, щодо обрання технологічного стеку для розробки веб-сайту юридично-психологічної допомоги для ветеранів, це є найкращими рішеннями використовувати для клієнтської частини Angular для серверної Node.js (express) та реляційну базу даних, таке поєднання технологій дозволить створити сучасний, продуктивний, легко підтримуваний та масштабований веб-сайт юридичної та психологічної допомоги з використанням штучного інтелекту для ветеранів. Аналіз технологій веб-розробки (див. табл. 2.2)

Таблиця 2.2 – Аналіз технологій веб-розробки

Джерело	Напрямок дослідження	Основні особливості	Практичне значення
PostgreSQL Documentation [11]	Реляційні бази даних	Надійність та стабільність	Зберігання даних
MySQL Documentation [12]	Системи управління БД	Робота зі структурованими даними	Організація БД
RiseUp Labs Guide [13]	Node.js розробка	Масштабованість систем	Backend-розробка
Angular and Node.js Full Stack [14]	Full-stack розробка	Взаємодія frontend/backend	Побудова веб-додатків
Modern Web Technology [15]	Сучасні веб-технології	Framework-based підхід	Веб-розробка
Backend Systems using Node.js [16]	Серверні системи	Оптимізація продуктивності	Масштабовані системи

### **2.3 Використання технологій штучного інтелекту у веб-сайті підтримки ветеранів**

У сучасних веб-сайтах все більше відіграють важливу роль технології штучного інтелекту, вони підвищують ефективність взаємодії між користувачем та програмним забезпеченням яке він використовує. Також особливо це корисно та актуально в тих веб-сайтах де йде різна онлайн підтримка користувачів у їх питаннях, що спрощує надання потрібної інформації, або ж консультування з якихось проблем чи питань, пов'язаних з цим веб-сайтом. В нашому веб-сайті юридично та психологічної допомоги, застосування технологій штучного інтелекту, значно автоматизує обробку запитів, питань від користувачів, також дозволить автоматизувати деякі процеси, що підвищить швидкість роботи веб-сайту та задоволеність користувачів від використання цього веб-сайту. Тому головна мета у використанні ШІ (штучний інтелект) це створення інтелектуального функціонального модуля, який здатний аналізувати запити, питання тощо від користувачів та надавати інформацію щодо них, або направляти до відповідних спеціалістів.

Функціонально штучний інтелект закладений для виконання декількох важливих задач, по перше це реалізація чату, в якому користувач зможе ставити свої питання, запити, та отримувати відповідні відповіді на питання та потрібну інформацію, без потреби звертатись із цим до відповідних спеціалістів. Наступне, це аналіз цих самих запитів, щоб веб-сайт міг віднести його до питань психологічної чи юридичної допомоги, щоб веб-сайт надалі міг надавати більш релевантні та коректні відповіді, та було розуміння в якому контексті йде розмова. За рахунок цих аналізів веб-сайт і надає відповіді на питання користувачів, якщо це типові питання, то надає з наявної бази питань, якщо ні, також може надати відповідь, але з рекомендацією звернутись до спеціаліста з цим питанням, тим самим і буде забезпечувати маршрутизацію звернень, якщо не виходить обробити запит, то буде направлено відповідне звернення до відповідного спеціаліста за

згоди користувача.

Сам процес обробки запитів від користувача можна представити у вигляді послідовності не складних дій, на першому етапі, користувач вводить свій запит до чату із штучним інтелектом, далі ШІ отримує запит та аналізує його, проводячи розбір тексту, можливо картинок, і віділяючи ключові фрази, слова, після чого визначає до якої категорії належить цей запит, чи то до психологічної допомоги, чи то до юридичної. Далі веб-сайт аналізує свою базу даних і якщо запит є типовим питанням, яке зустрічається постійно, та на нього вже є чітка відповідь, то він її надає, якщо ні то також надає відповідь базуючись на інформації з надійних джерел, але все одно рекомендує звернутись з цим питанням до спеціаліста. Таким чином у веб-сайті поєднано як і автоматизовану обробку так і можливість втручання людини для консультації з більш складних питань. Процес обробки запитів (див. рис. 2.4)

Ще до переліку основних переваг застосування штучного інтелекту у веб-сайті, це доступність веб-сайту 24 години на добу, тобто користувач може у будь який час зайти та скористатись веб-сайтом, отримавши відповіді на його запитання, консультування щодо деякої інформації, незалежно від того доступний відповідний спеціаліст у цей час чи ні. З цього випливає, що значно пришвидшується обробка всіх запитів, бо на більшість з них зможе відповісти штучний інтелект, що позитивно впливає на користувацький досвід, а також значно зменшить навантаження на відповідних спеціалістів, і вони зможуть більш швидко обробляти складні випадки та запитання, а також за рахунок аналізу запитів від штучного інтелекту, вдається більш точно класифікувати напрямок запиту, що також не мало важно у аналізі та обробці запитів від користувачів. Незважаючи на всі переваги, також треба враховувати деякі ризики та обмеження, по перше при аналізі складних запитів, за рахунок не повної бази відповідей та інформації, можливе надання не точної інформації, тому у таких випадках система буде рекомендувати звернутись до відповідного спеціаліста. Та головний момент це забезпечення конфіденційності користувачів та їх інформації, щоб ця інформація не попадала

далі в інтернет, а залишався конфіденційною між користувачем та історією його чату.

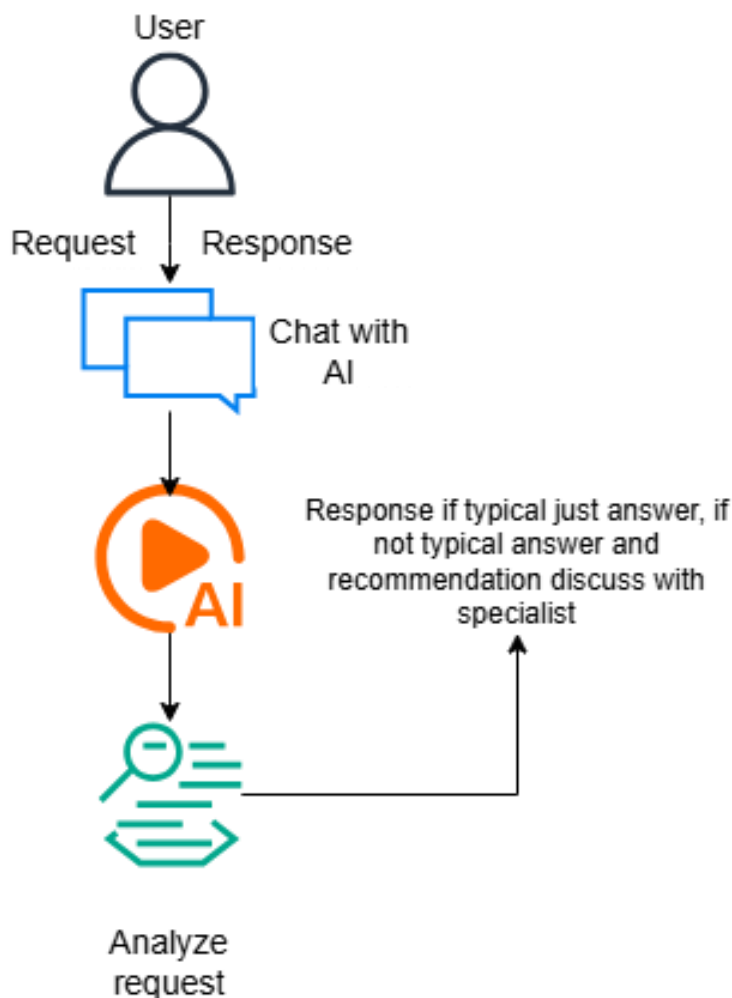


Рисунок 2.4 – Процес обробки запитів

У перспективі розвитку, є багато варіантів, тому що системи штучного інтелекту можуть використовувати все більш складні алгоритми аналізу, наповнювати свою базу знань, навчатись, та обробляти все більше запитів, надавати точніші відповіді на запити користувачів, також веб-сайт може індивідуально підлаштовуватись до поведінки користувача, враховувати історію звернень, та надавати все більш точні відповіді для цього користувача.

Виходячи з усього вище перерахованого та всіх плюсів а також недоліків штучного інтелекту, його використання є достатньо обгрунтованим та доцільним в

вею-сайті юридичної та психологічної допомоги для ветеранів. Аналіз штучного інтелекту та чат-ботів (див. табл. 2.3)

Таблиця 2.3 – Аналіз штучного інтелекту та чат-ботів

Джерело	Напрямок дослідження	Основні особливості	Практичне значення
IBM NLP [17]	Обробка природної мови	Аналіз текстових запитів	AI-консультації
OpenAI API [18]	Генеративний штучний інтелект	Формування відповідей	Чат-бот
Google ML Crash Course [19]	Машинне навчання	Побудова AI-моделей	Аналіз даних
Web-based Chatbots [20]	AI-системи підтримки	Автоматизація взаємодії	Онлайн-консультації
ChatGPT Security [21]	Безпека AI	Аналіз ризиків	Захист даних
Vulnerability Detection [22]	Аналіз вразливостей	Пошук проблем безпеки	Надійність систем
Chatbot Architecture [23]	Архітектура чат-ботів	Організація AI-модуля	Побудова чат-систем

#### 2.4 Забезпечення безпеки та надійності веб-сайту

Під час розробки веб-сайту, таке питання як забезпечення безпеки та надійності системи є одним із головних. Особливо це важливо у тих випадках коли веб-сайт працює з різними персональними даними які надають користувачі, особливо як у випадку нашого веб-сайту юридичної та психологічної допомоги з використанням штучного інтелекту для ветеранів. При роботі такого веб-сайту конфіденційність даних це один з головних моментів у всій роботі веб-сайту, щоб

не було витоку даних до інтернету або зловмисників. Це і передбачає продумана система безпеки яка не дає інформації витікати, поза межі веб-сайту, а також захист від зовнішніх та внутрішніх загроз. Натомість надійність веб-сайту це її можливість обробляти запити без збоїв, стабільно працювати 24 години на добу, та залишатись доступним.

Перший та базовий елемент який забезпечує безпеку при користування веб-сайтом, це автентифікація користувачів, що дозволяє підтвердити особу користувача, та відображати інформацію тільки власнику цієї інформації. У більшості випадків на сайтах це відбувається за допомогою ведення логіна та пароля. Після успішної автентифікації у системі, відбувається авторизація. Авторизація відповідає за визначення прав доступу на цьому веб-сайті, що ти можеш переглянути, що змінити та інше. Сучасні методи зазвичай використовують токени доступу такі як JSON Web Token (JWT). Після того як користувач увійшов, веб-сайт генерує унікальний токен, за рахунок якого є підтвердження особи цього користувача. Саме такий підхід дозволяє:

- уникнути зберігання сесій на сервері;
- забезпечити масштабованість веб-сайту;
- підвищити безпеку взаємодії клієнта та сервера;
- підвищити безпеку зберігання конфіденційних даних.

Процес автентифікації користувача (див. рис. 2.5)

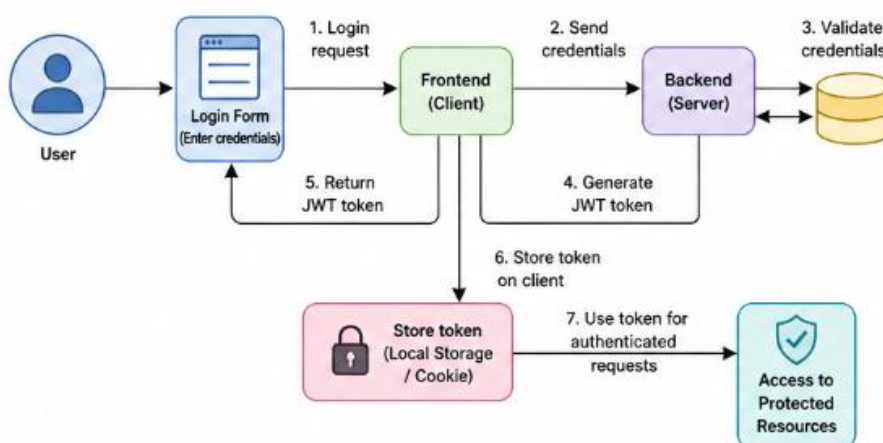


Рисунок 2.5 – Процес автентифікації користувача

Так як веб-сайт працює з персональними даними, потрібно забезпечити захист інформації на всіх етапах взаємодії з ним. Тому передача даних між клієнтом та сервером відбувається за протоколом HTTPS, це безпечна версія звичайного HTTP протоколу який забезпечує шифрування даних при передачі. Також увагу треба звернути на зберігання паролів, та використовувати особливі алгоритми шифрування та хешування використовуючи вже існуючі бібліотеки для цього, це захистить дані користувачів навіть при якомусь витокі інформації з бази даних. І також момент з налаштуванням контролю доступу, користувач має мати доступ лише до своїх даних та інформації, щоб інші користувачі не могли переглянути приватну інформацію інших користувачів. Не менш головним є забезпечення безпеки від зовнішніх веб-атак. З поширених можливих атак це може бути SQL-ін'єкція, тобто це коли якимись способами з сайту дозволяється робити сторонні запити до таблиць, та для запобігання цьому потрібно використовувати параметризовані запити. Ще є поширені два варіанти атак це XSS та CSFR, перший варіант атаки, це коли дозволяється виконання шкідливого коду у браузері, для цього як раз потрібно виконувати валідацію та очищення ведених даних. Другий варіант це виконання дій від імені якогось користувача, як раз для цього використовуються спеціальні токени.

Надійність веб-сайту, визначається його здатністю працювати без перебоїв 24 години на добу, навіть коли є підвищене навантаження на сайт. Для того щоб забезпечити таку роботу сайту, потрібно коректно обробляти помилки як і всередині веб-сайту та і показувати правильно їх користувачу, тобто щоб веб-сайт не просто падав у разі помилки, а правильно їх обробляла та завершувала свої процеси безпечно. Це дозволить не падати веб-сайту під час великого навантаження на нього і ще одним моментом є налаштування резервного копіювання даних, щоб навіть у випадку падіння веб-сайту, не втратити всі дані користувачів, та дуже швидко відновити роботу системи. Також для забезпечення безперервної роботи потрібно щоб відбувався постійний моніторинг веб-сайту, це потрібно для того щоб вчасно відслідковувати слабкі місця при навантаженнях веб-

сайту, знати що треба покращити для кращої продуктивності, та що переробити, в цьому також може допомогати штучний інтелект який буде автоматично відслідковувати усі дані.

Виходячи зі всього вищеперерахованого та сказаного, то забезпечення безпеки та надійності веб-сайту, є одним із ключових факторів, для успішності роботи усього веб-сайту, та довіри від користувачів про цілісність їх даних, та можливості звернутись 24 години на добу. Аналіз методів забезпечення систем (див. табл. 2.4)

Таблиця 2.4 – Аналіз методів забезпечення безпеки веб-сайтів

Джерело	Напрямок дослідження	Основні особливості	Практичне значення
OWASP Top 10 [24]	Веб-безпека	Аналіз сучасних загроз	Захист системи
Web Security Guidelines [25]	Захист веб-додатків	Security policies	Безпека даних
SQL Injection [26]	Захист баз даних	Параметризовані запити	Безпечна робота з БД
Cross-Site Scripting [27]	Захист frontend	Валідація введених даних	Безпечний інтерфейс
Cross-Site Request Forgery [28]	Захист HTTP-запитів	CSRF token	Захист взаємодії
HTTP Overview [29]	HTTP-протокол	Передача даних	Клієнт-серверна взаємодія
JSON Guide [30]	Формат JSON	Структурування даних	REST API
HTTPS [31]	Шифрування даних	SSL/TLS	Захищена передача
JWT [32]	Авторизація користувачів	Token-based auth	Контроль доступу

---

## Висновки до розділу 2

У другому розділі було проведено аналіз та обґрунтування технологій для розробки веб-сайту юридичної та психологічної допомоги для ветеранів з використанням штучного інтелекту. За результатами аналізу, було вирішено використовувати клієнт-серверну архітектуру, яка забезпечує сучасний та зручний підхід для розробки такого веб-сайту. Також були обрані та обґрунтовані доцільність використання технологій для клієнтської частини це Angular, для серверної частини це Node.js, для бази даних це реляційна база даних MySQL, а також розглянули чому важлива безпека та надійність веб-сайту, тому використання саме цих технологій дозволить розробити легко підтримуваний, легко масштабований, продуктивний веб-сайт який буде задовільняти потреби користувачів.

Таким чином було сформовано технологічну основу для подальшої розробки веб-сайту, що дозволяє перейти до етапу її проектування та реалізації.

## **3 ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБ\_САЙТУ ЮРИДИЧНОЇ ТА ПСИХОЛОГІЧНОЇ ДОПОМОГИ З ВИКОРИСТАННЯМ ШІ ДЛЯ ВЕТЕРАНІВ**

### **3.1 Проектування структури веб-сайту**

Під час проектування веб-сайту юридичної та психологічної допомоги з використанням ШІ для ветеранів, основна увага була на тому щоб створити зручний, легко масштабований та зрозумілий по структурі веб-сайт який буде забезпечувати стабільну взаємодію між клієнтом та сервером. Також беручи до уваги специфіку роботи веб-сайту та взаємодію з персональними даними користувачів, веб-сайт повинен забезпечувати також і захист даних користувачів, а також надійність та можливість розширення функціоналу.

Основна структура системи розробляється за тим принципом що розподіляє функціонал між окремими компонентами. При такому підході до розробки це дозволяє спростити підтримку веб-сайту, забезпечити легкість та зручність до змін та впровадження нової функціональності на сайті. До основних клієнтів веб-сайту відносяться клієнтська частина, серверна частина, база даних та модуль штучного інтелекту.

Користувач взаємодіє з веб-сайтом через клієнтський інтерфейс користувача, що надає доступ до основних функцій та можливостей веб-сайту. За допомогою інтерфейсу у користувача є можливість проходити авторизацію, переглядати інформаційні матеріали, надсилати звернення, отримувати консультаційну підтримку від спеціалістів, а також головне спілкуватись з чат-ботом на основі штучного інтелекту, тому основне завдання клієнтського інтерфейсу це забезпечити зручне користування веб-сайтом.

Серверна частина веб-сайту відповідальна за обробку HTTP запитів зі сторони клієнта до серверу, обробки основної бізнес логіки веб-сайту, взаємодію з базою даних, тому на сервері розробляється логіка авторизації користувачів, обробка їх запитів та повідомлень, та взаємодія зі штучним інтелектом для обробки

ним запитів.

Для того щоб зберігати дані користувачів отримані на нашому веб-сайті та відправлені на сервер, викорисовується база даних, у якій передбачено зберігання різної інформації користувачів від їх даних до запитів та звернень, та база даних проєктується таким чином щоб легко підтримувати великі обсяги інформації та забезпечувати надійні зв'язки.

Також одним з головних компонентів веб-сайту є штучний інтелект, він використовується для автоматизації частини процесів наприклад консультаційних підтримки з юридичних питань чи психологічної допомоги користувачеві. Тобто штучний інтелект забезпечує обробку текстових повідомлень від користувача та формує для нього відповіді на основі заданої йому логіки роботи. Тому використання штучного інтелекту автоматизує процес взаємодії з користувачем та зменшує навантаження на відповідних спеціалістів. Структура веб-сайту (див. Рис. 3.1).

Також під час розробки веб-сайту було приділено увагу необхідності розділення функціоналу між окремими модулями веб-сайту. Такий підхід до розробки дозволяє організувати більш зручну структуру коду, що спрощує подальшу підтримку та масштабування веб-сайту. У межах веб-сайту передбачено використання окремих функціональних модулів, і кожен з яких відповідає за виконання певних задач.

Один з таких модулів це авторизація та автентифікація користувача, цей модуль забезпечує реєстрацію, перевірку облікових записів, права доступу до різного функціоналу веб-сайту. Використання модуля авторизації забезпечить захист персональних даних користувачів а також обмежить доступ до деякого функціоналу наприклад як функціонал спеціаліста чи адміна.

Ще одним модулем веб-сайту є обробка звернень користувача, цей модуль розроблений для створення, зберігання та обробки цих запитів від користувачів, пов'язаних будь то юридична допомога чи психологічна. Цим самим передбачено збереження історій звернень та подальший аналіз цих запитів.

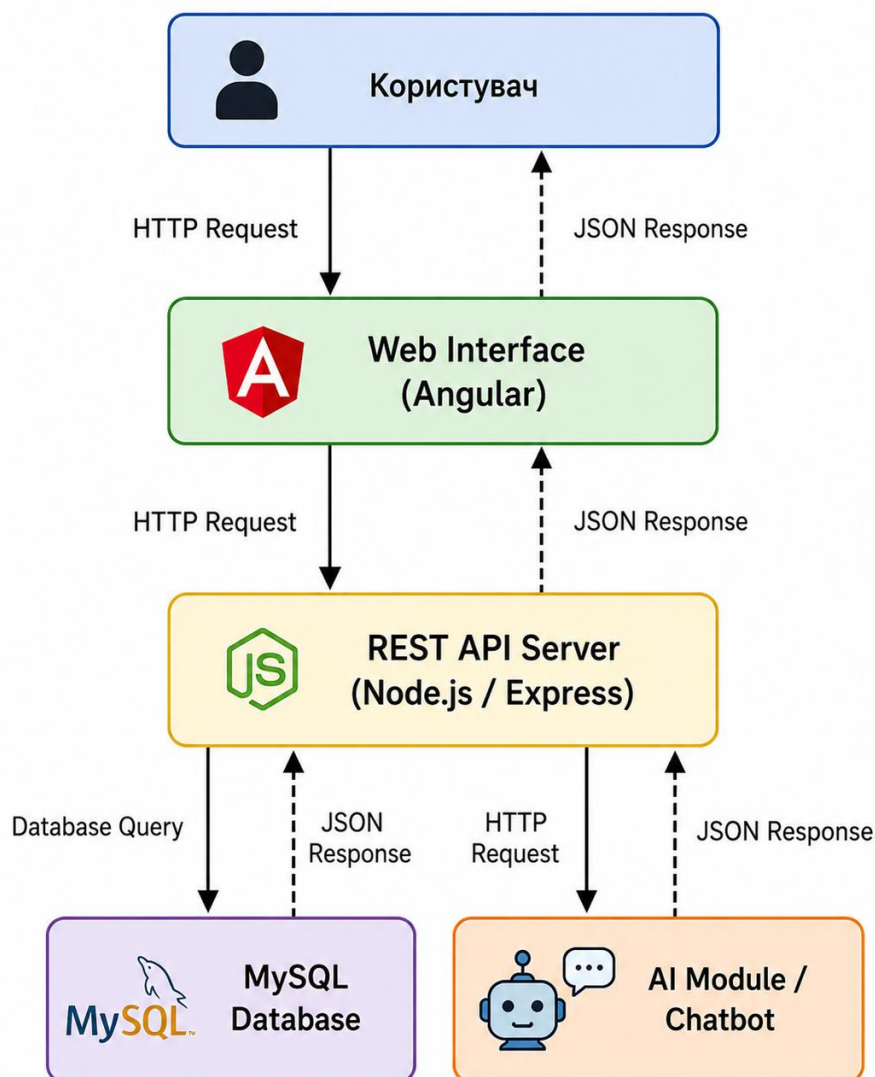


Рисунок 3.1 – Структура веб-сайту

Також одним з головних модулів веб-сайту це взаємодія із чат-ботом на основі штучного інтелекту, який дає змогу автоматично обробляти запити користувачів, формувати відповіді. Користувач через інтерфейс користувача надсилає запит після чого він надходить на сервер та передає на AI модуль для обробки та надання відповіді.

Було і враховано подальшу зручність підтримки та масштабування веб-сайту тому кожен із модулів розробляється окремо та незалежно від інших для того щоб у майбутньому було прстіше змінювати або модернізувати веб-сайт без

великозатратних змін всієї структури. Таким чином і було враховано безпеку та надійність системи за рахунок окремих модулів простіше забезпечити безпеку даних та безперебійну роботу веб-сайту. Тому запропонована структура веб-сайту забезпечує зручну організацію коду, модулів та подальші можливості для розвитку та підтримки веб-сайту.

### **3.2 Проектування функціональної структури веб-сайту**

Важливим етапом проектування веб-сайту юридичної та психологічної допомоги підтримки ветеранів з використанням ШІ є визначення функціональної структури веб-сайту та її основних можливостей для різних категорій користувачів. Функціональна структура зазначає набір можливих дій з які можуть виконувати користувачі і забезпечує організацію взаємодії між окремими модулями веб-сайту.

Під час розробки функціональної структури веб-сайту було приділено велику увагу користувацькому досвіду роботи з інтерфейсом а тобто розробити максимально простий та доступний інтерфейс для взаємодії і отримання юридичної та психологічної допомоги, і також враховано наявність декількох категорій користувачів для кожної з яких передбачено свій набір функціональних можливостей для роботи з веб-сайтом.

Основна категорія користувачів це є ветерани які прийшли на веб-сайт отримати юридичну або психологічну допомогу, за допомогою інтерфейсу вони мають змогу проходити авторизацію на веб-сайті, переглядати різні інформаційні матеріали, створювати звернення до відповідних спеціалістів а також спілкуватись з чат-ботом на основі ШІ і отримувати консультаційну допомогу.

Окремою категорією також є адміністратори веб-сайту та відповідні фахівці з питань юридичної чи психологічної допомоги, адміністратори та відповідно фахівці мають розширений доступ до веб-сайту ніж звичайні користувачі для того щоб мати можливість переглядати звернення від користувачів, обробляти, оновлювати відповідну інформацію та інше. Тому функціональні можливості адміністратора та фахівця на веб-сайті є дуже подібними одна до одної, оскільки і

ті і ті працюють із запитами та забезпечують підтримку користувачів, а також адміністратори займаються слідкуванням за контролем роботи всього веб-сайту та можуть виправляти деякі неполадки на сайті пов'язані з помилками у текстах матеріалів або тощо.

В загалом для користувачів веб-сайта передбачаються реалізація таких функціональних можливостей:

- реєстрація та авторизація;
- перегляд інформаційних матеріалів;
- створення звернення;
- взаємодія із чат-ботом;
- отримання консультацій;
- перегляд історії звернень;
- редагування персональної інформації.

Для адміністраторів та фахівців передбачається можливість:

- перегляду звернень користувачів;
- обробки повідомлень;
- надання консультацій;
- перегляду історій звернення;
- взаємодії із користувачами;
- управління інформаційними матеріалами;
- контролю роботи системи.

Функціональні можливості користувачів (див. рис. 3.2)

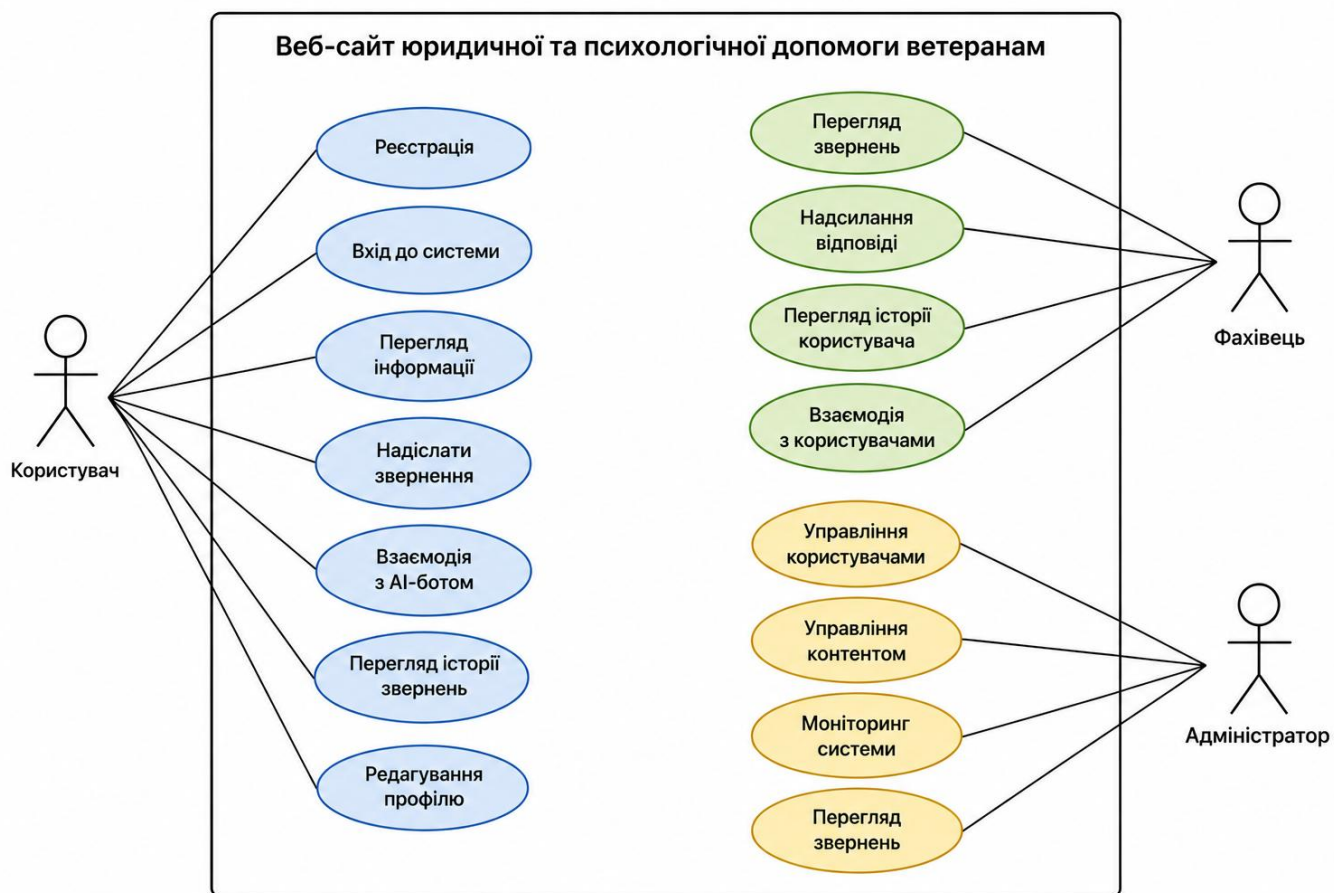


Рисунок 3.2 – Функціональні можливості користувачів веб-сайту

Під час проектування функціональної структури також враховувалась необхідність забезпечити простого та зрозумілого інтерфейсу зі зрозумілою навігацією по сайту. Тобто інтерфейс має забезпечувати швидкий та зрозумілий доступ до всіх основних функцій, матеріалів не створюючи незручностей під використання веб-сайту користувачем.

Одна з ключових функцій веб-сайту є взаємодія користувача з інтерфейсом чат-боту на основі штучного інтелекту. Передбачено такі можливості як надсилання повідомлень та отримання автоматичних відповідей від штучного інтелекту. Чат-бот використовується для надання базових відповідей та допомоги з навігацією по сайту, при отриманні незвичайного запиту також буде надана відповідь але з порадою для уточнення звернутись до кваліфікованого відповідного

спеціаліста чи то для юридичної допомоги чи психологічної в залежності від запиту користувача.

Також важливим є модуль обробки звернень, цей модуль забезпечує створення зверень користувачем, їх зберігання та подальший перегляд для відповідних фахівців щоб надати необхідну допомогу користувачеві за його зверненням до спеціалістів. Основні функціональні можливості веб-сайту (див. табл. 3.1)

Таблиця 3.1 – Основні функціональні можливості веб-сайту

Функціональні можливості	Опис
Реєстрація користувача	Створення облікового запису
Авторизація	Вхід до веб-сайту
Робота з чат-ботом	Отримання консультаційної підтримки
Створення звернень	Надсилання запиту через веб-сайт
Перегляд історії звернень	Доступ до попередніх запитів
Редагування профілю	Оновлення персональної інформації
Обробка звернень	Робота фахівців із запитами
Управління контентом	Оновлення інформаційних матеріалів
Моніторинг системи	Контроль роботи вебзастосунку

Також було враховано такі питання під час розробки функціональної структури веб-сайту як питання масштабованості веб-сайту, тобто його можливість для подальшого розвику та модернізації без потреби вносити великі зміни вже до

існуючого та працюючого застосунку. Було враховано зручну взаємодію клієнтської частини та серверної, для цього дії користувачів реалізуються через API-запити до серверної частини, що централізує всі запити та відповіді і забезпечує більш зручну організацію роботи веб-сайту. Ще наступним питанням функціональної структури було враховано безпеку захисту персональних даних користувачів, щоб користувачі не переживали за їх персональні дані які вони вносять на сайт або пишуть у чат-бот тим самим передбачено авторизації користувачів на сайті.

Виходячи з усього вище перерахованого функціональна структура веб-сайту забезпечує реалізацію основних можливостей веб-сайту, підтримку взаємодії між користувачем та фахівцями, а найголовніше створює стійку основу для подальшого розвитку веб-сайту.

### **3.3 Проектування структури бази даних веб-сайту**

Один із важливих етапів розробки веб-сайту юридичної та психологічної допомоги ветеранам з використанням ШІ є розробка структури бази даних. База даних забезпечує зберігання інформації, яка необхідна для повного функціонування веб-сайту. Під час розробки структури бази даних головна увага була приділена тому що забезпечити цілісність, зручності даних та можливості подальшого розширення роботи веб-сайту. Структура бази даних має забезпечувати ефективну роботу із даними користувачів, їх зверненнями, повідомленнями та результатами взаємодії.

Для розробки веб-сайту було обрано реляційну базу даних MySQL, яка забезпечує підтримку зв'язків між таблицями, структуроване зберігання інформації та можливість виконання SQL-запитів. Використання саме реляційної бази даних полегшує зв'язок між сервером та базою даних, та робить зручну організацію даних.

Під час розробки бази даних було визначено основні сутності системи:

- користувачі;
- звернення;
- повідомлення;
- фахівці;
- адміністратори;
- історія взаємодії із чат-ботом;

Кожна із сутностей представлена окремою таблицею всередині бази даних та має відповідний набір полів, які потрібні для зберігання інформації. Між цими таблицями побудовані зв'язки які дозволяють забезпечити їх взаємодію.

Основна таблиця бази даних це таблиця користувачів, у якій зберігаються дані про користувачів веб-сайту які пройшли на ньому реєстрацію. У таблиці розміщено персональні дані користувачів, інформацію для авторизації та деякі параметри які потрібні для ідентифікації (див. табл. 3.2)

Таблиця 3.2 – Основні поля таблиці користувачів

Поле	Тип даних	Опис
Id	INT	Унікальні ідентифікатор
full_name	VARCHAR	ПІБ користувача
email	VARCHAR	Електронна пошта
password	VARCHAR	Хешований пароль
role	VARCHAR	Роль користувача
created_at	DATETIME	Дата реєстрації

Для того щоб працювати із запитами користувачів також використовується таблиця запитів, де зберігаються самі запити, їх стастути та результати роботи з ними (див. табл. 3.3)

Таблиця 3.3 – Основні поля таблиці запитів

Поле	Тип даних	Опис
id	INT	Ідентифікатор запиту
user_id	INT	Ідентифікатор користувача
category	VARCHAR	Категорія запиту
message	TEXT	Текст запиту
status	VARCHAR	Статус обробки
created_at	DATETIME	Дата створення

Також передбачено таблицю для зберігання історії взаємодії тобто повідомлень між користувачем, фахівцем або чат-ботом.

Таблиця 3.4 – Основні поля таблиця повідомлень

Поле	Тип даних	Опис
id	INT	Ідентифікатор повідомлення
sender_id	INT	Відправник повідомлення
request_id	INT	Ідентифікатор запиту
message_text	TEXT	Текст повідомлення
created_at	DATETIME	Час надсилання

Як можна побачити у цих таблицях під час розробки структури бази даних також було враховано зв'язок між таблицями як видно таблиця користувачів і запитів пов'язані через `user_id`, щоб вони могли працювати у парі а таблиця запитів з таблицею повідомлень через `request_id`. Це дає забезпечення цілісності даних та підтримку зв'язку між ними. Також можна роглянути діаграму зв'язків (див. рис. 3.3)

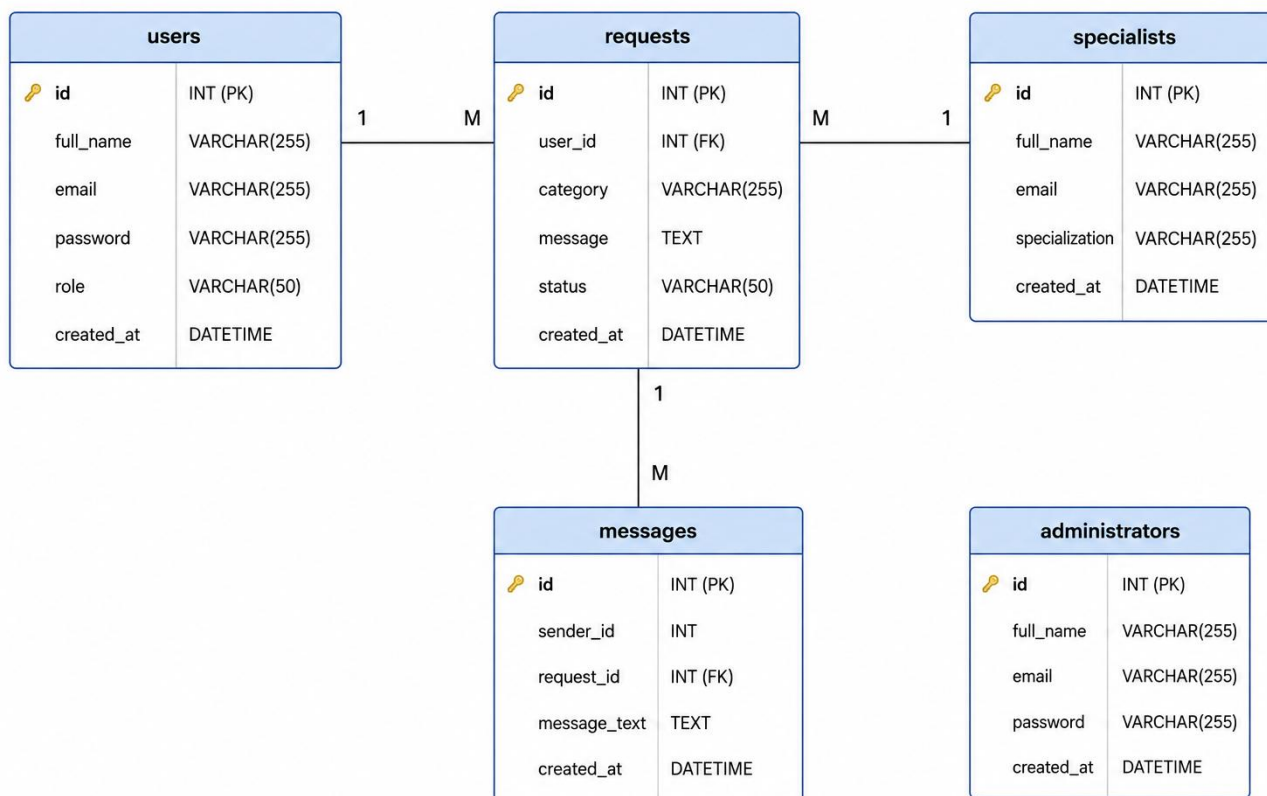


Рисунок 3.3 – Діаграма зв'язків

Також було враховано безпеку бази даних тому для зберігання важливої інформації таких наприклад як паролі використовуються хешування а для отримання іншої інформації модуль авторизації користувачів. Наступним важливим було враховано можливості розширення та масштабованості бази даних для того щоб при додаванні нових таблиць не потрібно було змінювати структуру

бази даних, а також для зручної взаємодії із сервером застосовуються SQL запити.

Виходячи з усього вище перерахованого запропонована структура бази даних дає нам зручне зберігання інформації, зручної взаємодії між компонентами та дає основу для подальшого розширення та реалізації нових функцій для веб-сайту юридичної та психологічної допомоги ветеранам з використанням ШІ.

### **3.4 Розробка клієнтської частини та інтерфейсу веб-сайту**

Інтерфейс користувача та цілком клієнтська частина веб-сайту є одним з основних компонентів, тому що через неї як раз користувач і взаємодіє з функціоналом веб-сайту. Під час розробки було приділено увагу для створення сучасного, адаптивного та головне легко зрозумілого інтерфейсу для користування, та отримання юридичної чи психологічної допомоги.

Для розробки клієнтської частини використовувався фреймворк Angular, який дає змогу створювати та розробляти SPA-застосунки, які є сучасними та зручними у використанні, тому що не потребують постійного перезавантаження сторінки, використовуючи динамічне оновлення інтерфейсу, це впливає як і на швидкість роботи інтерфейсу так і на позитивний досвід користувача при використанні сайту.

Було використано компонентний підхід який є основоположним для Angular додатків, де компоненти реалізуються кожен незалежно та можуть бути перевикористані у разі потреби, що спрощує підтримку та розширення системи у разі потреби.

Структура Angular-проекту реалізована за модульним принципом, тобто основні компоненти, сторінки, сервіси розміщені у відповідних директоріях всередині самого проекту (див. рис. 3.4)

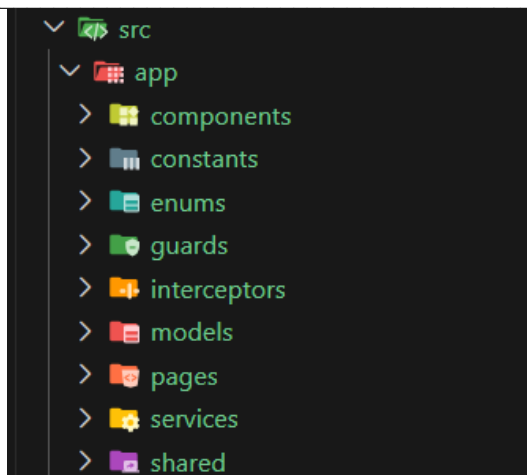


Рисунок 3.4 – Структура клієнтської частини Angular проекту

У папці `components` розроблено різні компоненти інтерфейсу окремо один від одного, до них входять головна сторінка, форма авторизації та реєстрації, сторінки роботи зі зверненнями та спілкування в чат-боті з використанням ШІ. У папці `pages` основні сторінки веб-сайту:

- головна сторінка;
- сторінка авторизації;
- сторінка реєстрації;
- профіль користувача;
- сторінка звернень;
- AI-чат;
- адміністративна панель.

Для налаштування навігації між сторінками веб-сайту використані вбудовані механізми `Angular Router` та `Angular guard`, які надають швидке переміщення між сторінками без потреби перезавантаження сторінки, а також контроль доступу до окремих сторінок. Схеми маршрутизації (див. рис. 3.5)

Для зв'язку з сервером з клієнтської частини веб-сайту використані `Angular` сервіси та `HTTP` запити до `REST API`. Через них реалізовано авторизацію користувача, роботу зі зверненнями, отримання необхідної інформації, взаємодія з чат-ботом.

```
export const routes: Routes = [
  { path: '', component: Home },
  { path: 'login', component: Login },
  { path: 'register', component: Register },
  {
    path: 'profile',
    component: Profile,
    canActivate: [AuthGuard],
  },
  {
    path: 'requests',
    component: Requests,
    canActivate: [AuthGuard],
  },
  {
    path: 'requests/:id',
    component: RequestDetails,
    canActivate: [AuthGuard],
  },
  {
    path: 'chat',
    component: Chat,
    canActivate: [AuthGuard],
  },
  {
    path: 'specialist',
    component: SpecialistDashboard,
    canActivate: [AuthGuard, RoleGuard],
    data: { roles: [UserRole.Specialist, UserRole.Administrator] },
  },
  {
    path: 'admin',
    component: Admin,
    canActivate: [AuthGuard, RoleGuard],
    data: { roles: [UserRole.Administrator] },
  },
  { path: '**', redirectTo: '' },
];
```

Рисунок 3.5 – Схеми маршрутизації

Також до авторизації було реалізовано систему механізму автентифікації за допомогою JWT токенів, після авторизації з серверу на клієнт надсилається унікальний токен який є ідентифікатором користувача та веб-сайт перевіряє за ним які доступи є у користувача до яких сторінок. Реалізовано декілька ролей користувачів на веб-сайті:

- користувач;
- фахівець;
- адміністратор.

В залежності від ролі, користувач має різні доступи до системи наприклад фахівець окрім основного функціоналу який доступний для звичайного користувача має ще

панель роботи зі зверненням користувачів, а адміністратор окрім цього також має доступ до панелі керування користувачами, та веб-сайтом.

Так як один з головних функціоналів веб-сайту це робота зі зверненнями, то користувач має можливість створювати ці запити, переглядати історії своїх запитів та отримувати відповіді від фахівців, для цього на веб-сайті розроблено окрема сторінка для створення та перегляду своїх запитів (див. рис. 3.6)

## Запити на підтримку

Створюйте правові, психологічні або технічні запити та відстежуйте їх виконання.

### Новий запит на підтримку

Коротко опишіть суть звернення та ключові деталі для швидшого призначення спеціаліста.

Назва звернення

Потрібна допомога з пенсійними документами

Категорія підтримки

Правова підтримка

Опис

Опишіть вашу ситуацію детально...

Надіслати запит

### Історія запитів

Назва	Категорія	Статус	Створено	
Як отримати житло ветерану від держави?	Правова підтримка	В роботі	May 26, 2026	<a href="#">Деталі</a>
Тяжко морально не можу з цим впоратись	Психологічна підтримка	Звершено	May 26, 2026	<a href="#">Деталі</a>
Потрібна допомога з оформленням документів для пільг	Правова підтримка	В роботі	May 26, 2026	<a href="#">Деталі</a>

Рисунок 3.6 – Сторінка роботи із запитами

Другий головний функціонал веб-сайту це чат-бот для допомоги користувачам на базі штучного інтелекту. Реалізовано окремий інтерфейс для взаємодії з ним де можна надсилати повідомлення, переглядати історію повідомлень та чату, і отримувати відповіді від штучного інтелекту з приводу

## питань користувача. Інтерфейс чат-боту з ШІ (див.рис. 3.7)

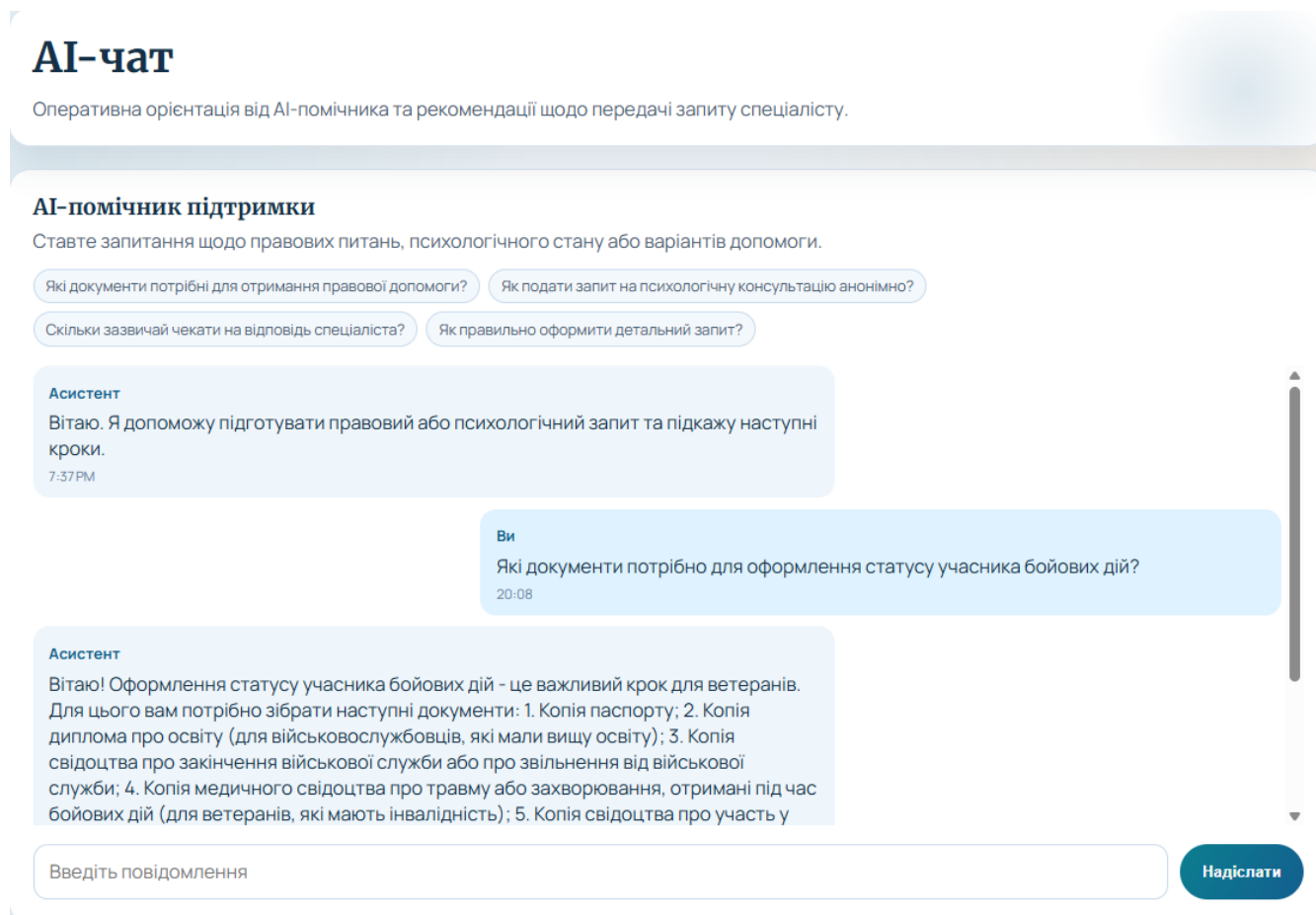


Рисунок 3.7 – Інтерфейс чат-боту з ШІ

Крім цього на веб-сайті реалізовано адміністративну панель та панель фахівця, які дозволяють переглядати запити користувачів, змінювати статуси, надавати відповіді та виконувати модерацію системи. Панель адміністратора веб-сайту (див.рис. 3.8)

Для стилізації веб-сайту використовувались технології SCSS, яка дозволяє організувати структуру стилів веб-сайту та забезпечувати адаптивність інтерфейсу для різних типів пристроїв. Також враховувались принципи UX/UI-дизайну під час розробки:

- зрозуміла структура сторінок;
- зручна навігація;

- адаптивність інтерфейсу;
- мінімізація складності взаємодії користувача з веб-сайтом.

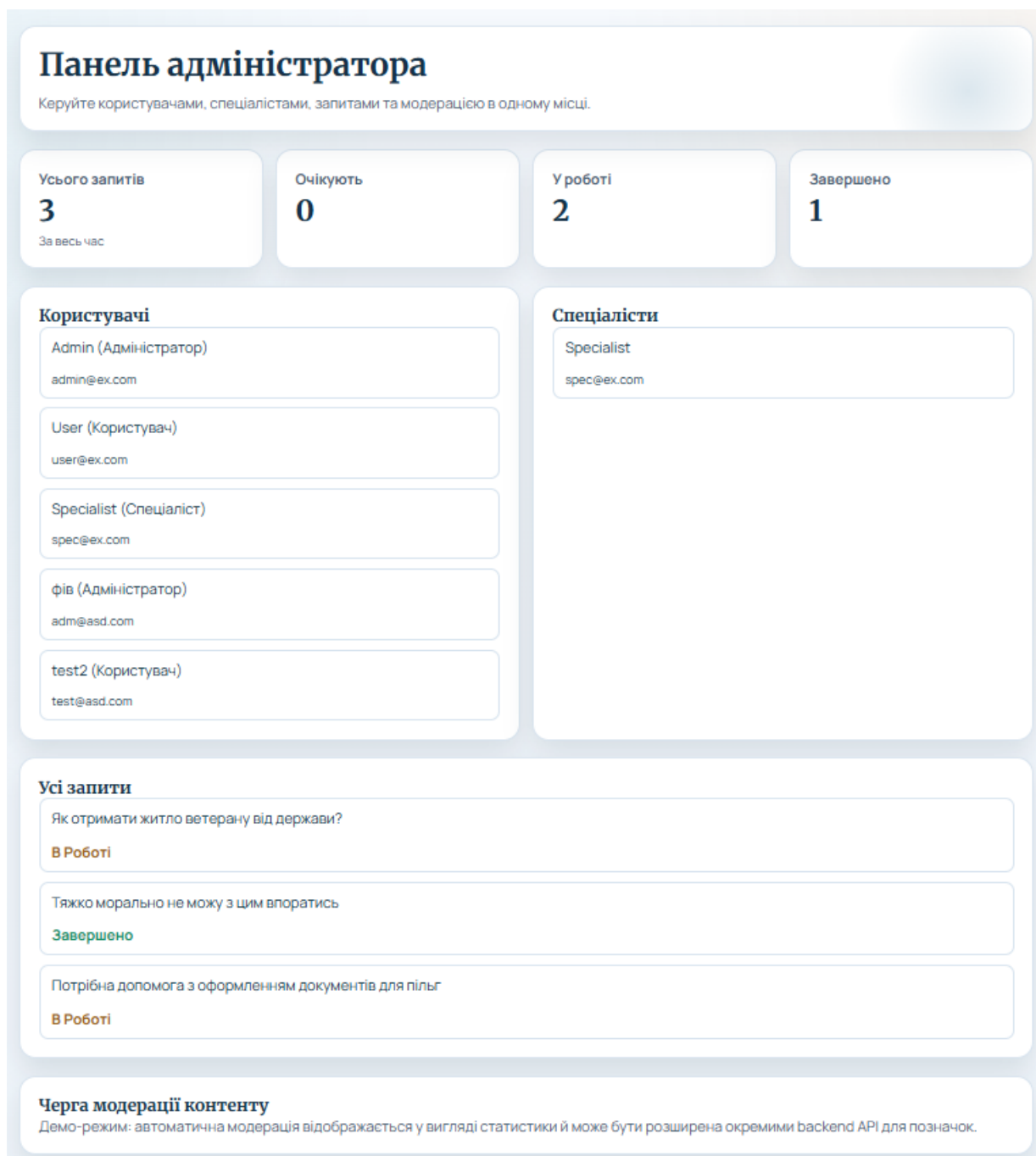


Рисунок 3.8 – Панель адміністратора

Також головна сторінка веб-сайту розроблена для зручності та націлена на те щоб при заході користувача на веб-сайт було відразу зрозуміло цільову тему веб-

сайті, зручність в логіні та реєстрації, виведено на головну, та основні теми за якими користувач може звернутись. Головна сторінка (див. рис. 3.9)

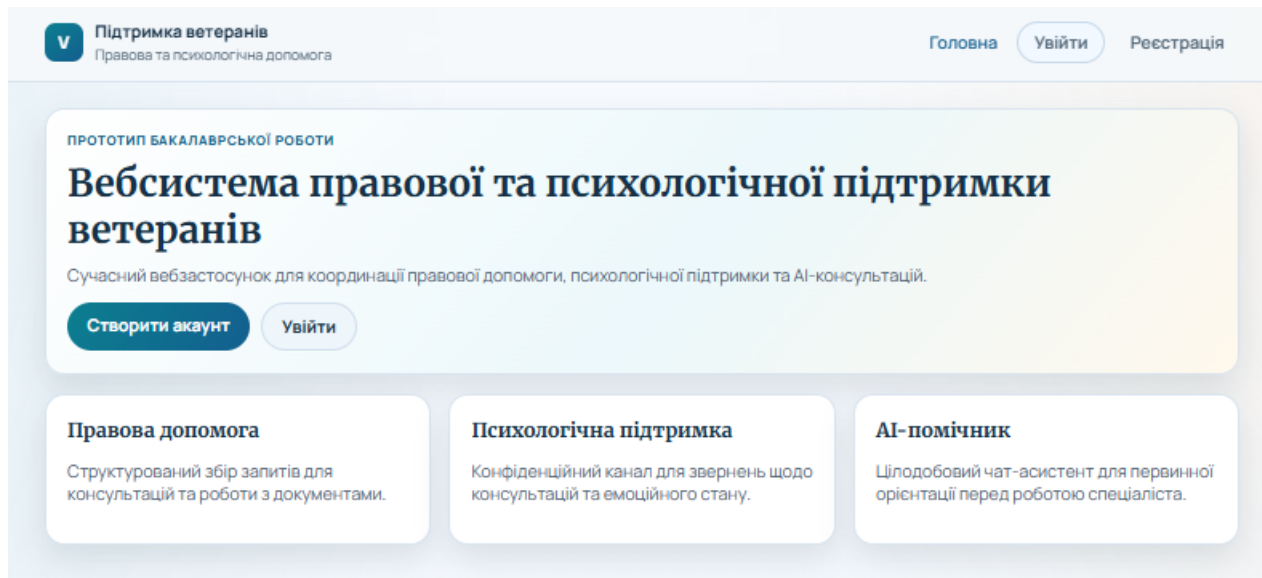


Рисунок 3.9 – Головна сторінка веб-сайту

Таким чином, за результатом розробки клієнтської частини та інтерфейсу було розроблено та створено сучасний SPA веб-сайт на основі Angular, який надає зручний інтерфейс користувача, підтримує взаємодію із серверною частиною через REST API, та реалізує основні функціональні можливості веб-сайту юридичної та психологічної допомоги ветеранам з використанням ШІ.

### 3.5 Розробка серверної частини та бази даних

Серверна частина займається обробкою HTTP-запитів, зв'язок з базою даних, реалізацію основної бізнес-логіки веб-сайту, також взаємодія клієнтської частини та ШІ-модуля. Під час розробки основну увагу було приділено модульності та структурі серверної частини, забезпечення безпеки даних користувача, реалізацію REST API для взаємодії із клієнтською частиною. Було використано для розробки технологію та платформу Node.js та фреймворк для нього Express.js. Використання цих технологій дозволяє ефективно обробляти велику кількість HTTP запитів та забезпечує асинхронну взаємодію із базою даних та клієнтською частиною, що

довозляє одночасно виконувати обробку багатьох запитів, без зупинки роботи веб-сайту та без погіршення швидкості виконання цих запитів, та за рахунок фреймворку спрощується реалізація маршрутів, та інших функцій.

Структура серверної частини веб-сайту організована за модульним принципом, окремо виділено маршрути, контролери, сервіси, та модулі роботи із базою даних. Структура серверної частини веб-сайту (див.рис. 3.10)

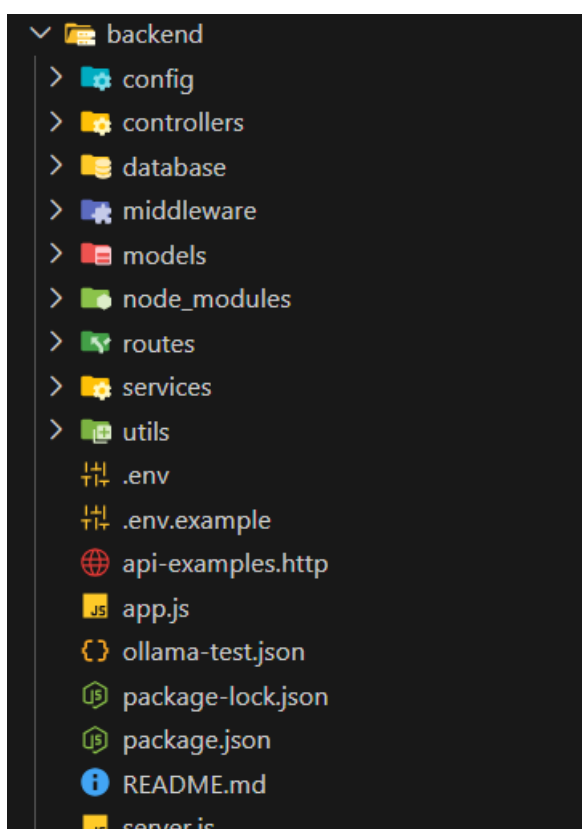


Рисунок 3.10 – Структура серверної частини веб-сайту

У директорії routes розроблено маршрути REST API, які обробляють HTTP-запити, контролери маю основну логіку обробки запитів, а сервіси забезпечують взаємодію із базою даних та ШІ. Через REST API реалізовано зв'язок між серверною та клієнтською частиною а саме:

- авторизацію та реєстрацію користувачів;
- роботу зі зверненнями;
- взаємодію з чат-ботом;

- перегляд інформації про користувача;
- адміністративне керування веб-сайтом.

Один з елементів серверної частини є автентифікація користувачів із забезпеченням безпеки входу використовуючи систему JWT токенів, що після авторизація на клієнт надходить унікальний токен який визначає користувача та його доступи на веб-сайті. За рахунок цього також розроблено рольову модель доступу та передбачено три основні ролі:

- користувач;
- фахівець;
- адміністратор.

Залежно від ролі сервер обмежує певні доступи до певного функціоналу веб-сайту.

Для бази даних було обрано реляційну базу даних MySQL, вона забезпечує впорядковане зберігання інформації про користувачів, запитів, взаємодій з чат-ботом. Основним таблицями в базі даних є:

- users;
- request;
- messages;

Таблиця users використовується для зберігання інформації про самих користувачів, їх ролі на веб-сайті, дані для авторизації користувачів. У таблиці requests зберігається інформація про запити користувачів, та статуси їх обробки. Таблиця messages використовується для того щоб зберігати інформації про повідомлення у чаті та історії взаємодії між користувачами та фахівцями.

Також коли розроблялися серверна частина в технічній частині коду було розроблено функція для перевірки JWT токенів, обробки різних помилок які можуть відбутися, щоб коректно їх зрозуміти та виправити, та контролю доступів на веб-сайті. Для обміну даними між сервером та клієнтом використано загальновизначений формат JSON, тому сервер повертає відповіді у структурованому форматі, які легко обробляти на клієнтській частині веб-сайту.

## Основні ендпоінти веб-сайту (див.табл. 3.5)

Таблиця 3.5 – Основні ендпоінти веб-сайту

Метод	Ендпоінт	Призначення
POST	/api/register	Реєстрація користувачів
POST	/api/login	Авторизація користувачів
GET	/api/profile	Отримання профілю
GET	/api/requests	Отримання звернень
POST	/api/requests	Створення звернення
POST	/api/chat	Взаємодія з ШІ

Для стабільної роботи сервера було забезпечено та розроблено централізовану обробку помилок, для того щоб правильно обробляти помилки які ми отримуємо та відразу розуміти в чому проблема. Крім цього сервер підтримує роботу з ШІ модулем веб-сайту, повідомлення з клієнтської частини надходить до серверу де він відправляє його до ШІ а потім надсилає на клієнт відповідь від ШІ.

Таким чином в результаті розробки серверної частини веб-сайту було розроблено сервер на основі Node.js та Express, з розробкою REST API маршрутів, який забезпечує взаємодію сервера та клієнта, ШІ модуля між сервером та клієнтом, взаємодію між базою даних сервером та клієнтом. Розроблена архітектура забезпечує масштабованість, модульність та безпеку веб-сайту юридичної та психологічної допомоги ветеранам з використанням ШІ.

### 3.6 Реалізація модуля штучного інтелекту

Одним з головних фішок функціоналу веб-сайту юридичної та психологічної допомоги ветеранам є модуль штучного інтелекту, який використовується для автоматизації допомоги користувачам та обробки деяких процесів, та надання консультацій через чат-бот. Використання ШІ дає змогу підвищити швидкість обробки звернень та надавати відповіді на типові питання користувачів, не

нагружаючи при цьому спеціалістів, що робить допомогу більш швидкою та автоматизованою. Під час розробки було розроблено окремий ШІ модуль який взаємодіє із серверною частиною веб-сайту через REST API. Основним призначенням ШІ модуля є:

- обробка повідомлень користувача;
- формування автоматичних відповідей;
- підтримка діалогової взаємодії;
- надання базових консультацій;
- перенаправлення складних звернень до фахівців.

Для розробки ШІ модуля, використана інтеграція з сервісом OpenAI ChatGPT через API. Взаємодія з ШІ відбувається через серверну частину веб-сайту, тобто через клієнтську частину надсилається повідомлення до сервера, він передає через API до ШІ а потім ШІ повертає згенеровану відповідь на сервер та сервер надсилає відповідь на клієнтську частину та там вже вона відображається користувачу. Схема взаємодії ШІ з веб-сайтом (див. рис. 3.11)

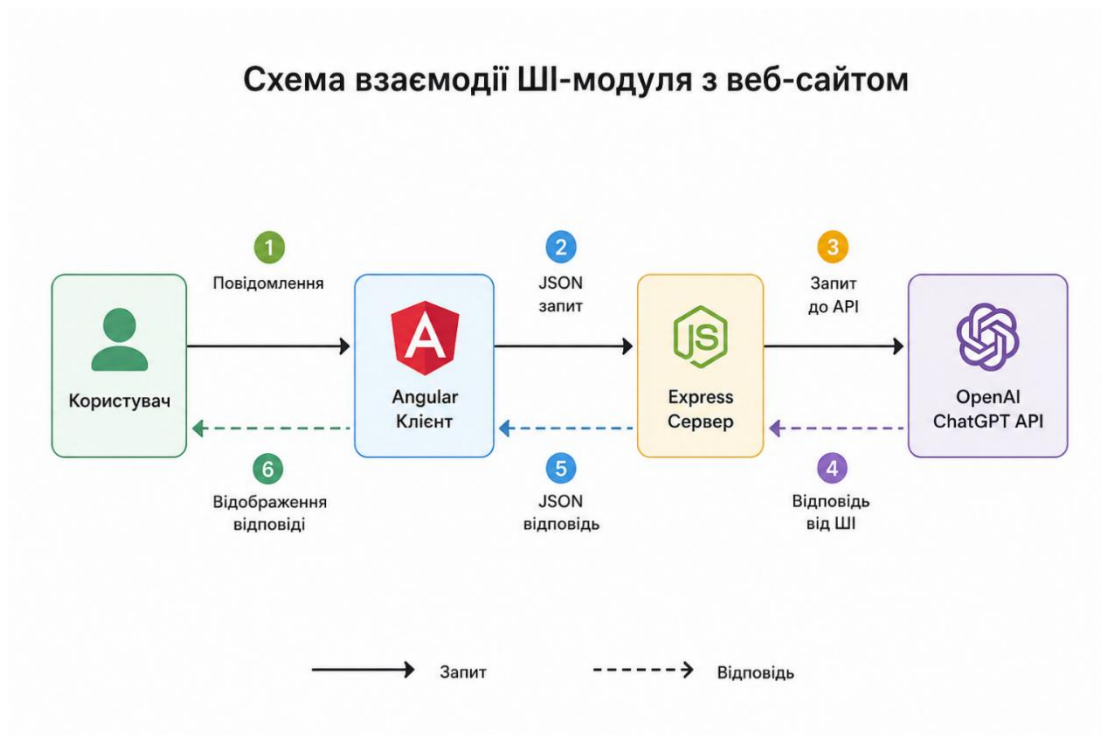


Рисунок 3.11 – Схема взаємодії ШІ з веб-сайтом

Сам принцип роботи ШІ полягає у послідовній обробці повідомлень користувача, після ведення повідомлення у чат-боті та надсилання повідомлення до сервера натиснувши відповідну кнопку надіслати, сервер обробляє отримані дані та передає текст повідомлення через API до ШІ, який вже генерує відповідь та надсилає її назад на сервер, після чого вже сервер формує JSON конструкцію та надсилає її на клієнтську частину. ШІ модуль використовується для:

- надання базових відповідей;
- допомоги користувачеві під час навігації по веб-сайту;
- відповідей на типові юридичні та психологічні запити;
- первинної консультації користувачів.

Також було враховано обмеження ШІ модуля, тобто у разі складних або специфічних запитів, ШІ може надати відповідь але з рекомендацією звернутись до фахівця з отримання коректної консультації, а у разі не можливості ШІ надати відповідь то відразу пропонує звернутись до фахівця.

Для того щоб забезпечити стабільну роботу ШІ модуля, сервер реалізує обробку помилок та перевірку правильності API запитів, щоб не відбувались збої на веб-сайті, у разі недоступності ШІ сервісу, веб-сайт повертає відповідне повідомлення про це користувачеві. Основні функції ШІ модуля (див. табл. 3.6)

Таблиця 3.6 – Основні функції ШІ модуля

Функція	Призначення
Обробка повідомлень	Аналіз тексту запиту користувача
Генерація відповідей	Формування відповіді від ШІ
Взаємодія через API	Обмін повідомленнями між клієнтом та сервером
Підтримка діалогу	Забезпечення чат-взаємодії
Первинна консультація	Надання базових рекомендацій

Також було враховано питання безпеки та конфіденційності даних

користувачів під час розробки та впровадження ШІ модуля. Передача повідомлень між клієнтом, сервером та ШІ відбувається через захищені HTTPS запити, а доступ до ШІ доступний лише авторизованим користувачам. Етапи обробки повідомлень ШІ (див. табл. 3.7)

Таблиця 3.7 – Етапи обробки повідомлень ШІ

Етап	Опис
1	Користувач водить повідомлення у чат та натискає надіслати
2	Клієнтська частина надсилає HTTPS – запит до серверу
3	Серверна частина обробляє повідомлення користувача
4	Сервер надсилає це повідомлення до OpenAI API(ШІ)
5	ШІ генерує текстову відповідь на повідомлення та надсилає на сервер
6	Сервер формує та надсилає відповідь у JSON форматі
7	Клієнтська частина відображає відповідь від ШІ користувачу

Виходячи з цього у результаті розробки ШІ модуля було створено автоматизовану підтримку користувачів на веб-сайті, яка надає генерацію відповідей у форматі чат-бота та підтримує взаємодію між користувачем, сервером та ШІ.

### 3.7 Аналіз результатів розробки

У результаті виконання роботи було розроблено веб-сайт юридичної та психологічної допомоги ветеранам з використанням ШІ, яка надає можливості та забезпечує взаємодію користувачів із відповідними фахівцями, надає можливість консультації з ШІ через чат-бот. Розроблений веб-сайт побудований з врахуванням всіх сучасних вебархітектур для забезпечення сучасності, підтримки та розвитку веб-сайту, а саме окремо клієнтська частина, серверна частина, база даних та ШІ модуль.

Також було розроблено основні функціональні можливості веб-сайту про які

було зазначено раніше у кваліфікаційній роботі, необхідні для роботи користувачів, фахівців та адміністраторів веб-сайту. Реалізовані функціональні можливості веб-сайту (див. табл. 3.8)

Таблиця 3.8 – Реалізовані функціональні можливості веб-сайту

Функціональні можливості	Опис реалізації
Реєстрація та авторизація	JWT – автентифікація користувачів
Робота із запитами	Створення та перегляд звернень
ІІІ-чат	Взаємодія користувача з ІІІ
Рольова модель	Підтримка ролей користувачів
Адміністративна панель	Керування користувачами та веб-сайтом
Панель фахівця	Робота зі зверненням користувачів
REST API	Взаємодія клієнтської та серверної частини
База даних MySQL	Збереження даних з веб-сайту

Також було враховано та забезпечено основні вимоги, сформовані на етапі постановки задачі. Розроблено сучасний та зручний SPA інтерфейс користувача, серверну частину, та ІІІ модуль підтримки користувачів. Відповідність реалізованого веб-сайту поставленим вимогам (див. табл. 3.9)

Таблиця 3.9 – Відповідність реалізованого веб-сайту поставленим вимогам

Вимога	Результат реалізації
Реєстрація користувачів	Реалізовано
Авторизація користувачів	Реалізовано
Створення запитів	Реалізовано
Обробка запитів	Реалізовано
Інтеграція з ІІІ	Реалізовано
Рольова модель доступу	Реалізовано
Адаптивний інтерфейс	Реалізовано

Головна перевага розробленого веб-сайту це використання сучасних вебтехнологій та архітектури які можуть бути легко підтримувані та масштабовані. Основні переваги розробленого веб-сайту (див. табл. 3.10)

Таблиця 3.10 – Основні переваги розробленого веб-сайту

Перевага	Характеристика
SPA архітектура	Швидка робота без перезавантаження сторінки
REST API	Зручна робота між компонентами веб-сайту
ШІ модуль	Автоматизована підтримка користувачів
Модульна структура	Спрощення підтримки та масштабування
JWT автентифікація	Захист доступу до веб-сайту
Адаптивний інтерфейс	Підтримка різних типів пристроїв

Тому за результати розробки було розроблено повноцінний веб-сайт юридичної та психологічної допомоги з використанням ШІ для ветеранів, який забезпечує легку його підтримку за рахунок сучасних технологій, та має переваги над іншими системами та веб-сайтами за рахунок своєї сучасності та простоти.

### Висновки до розділу 3

У третьому розділі кваліфікаційної роботи було виконано проектування та програмну реалізацію веб-сайту юридичної та психологічної допомоги з використанням ШІ для ветеранів Російсько-Української війни 2014-2015. Під час розробки було спроектовано архітектуру веб-сайту, його функціональні можливості та реалізовані основні вимоги до веб-сайту.

Було розроблено клієнтську структуру з використанням сучасних вебтехнологій таких як Angular з використанням SPA підходу, що дає швидку та зручну взаємодію з веб-сайтом, розроблено основні сторінки веб-сайту та функціонал який присутній на цих сторінках. Була реалізована серверна частина

також з використанням сучасних серверних технологій на основі Node.js та Express.js, серверна частина забезпечує обробку основної бізнес логік, HTTP запитів, автентифікації з використанням JWT токенів, зв'язок з базою даних MySQL, та взаємодії з ШІ модулем. ШІ модуль був реалізований на основі інтеграції з OpenAI ChatGPT API, що забезпечує автоматизацію процесів обробки запитів від користувачів. Крім цього виконано аналіз результатів розробки, де було обозначено вимоги які були раніше до веб-сайту та що було реалізовано, а також зазначені основні переваги веб-сайту та його можливостей.

Виходячи з цього при виконанні третього розділу, було розроблено повноцінний веб-сайт юридичної та психологічної допомоги з використанням ШІ для ветеранів Російсько-Української війни 2014-2025, що підлягає під всі вимоги сучасного веб-сайту.

## 4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ВЕБ-САЙТУ

### 4.1 Методики та функціональне тестування веб-сайту

Після завершення розробки веб-сайту юридичної та психологічної допомоги з використанням ШІ для ветеранів Російсько-Української війни 2014-2025 було зроблено функціональне тестування веб-сайту, для виявлення можливих проблем, чи коректно працює веб-сайт та відповідність реалізованого функціоналу до вимог. Під час тестування було перевірено клієнтську частину, серверну частину, базу даних та ШІ модуль. Особливо було приділено увагу коректності взаємодії клієнтської та серверної частини. Для перевірки функціональних можливостей веб-сайту було використано метод функціонального тестування, який дає можливість оцінити правильність роботи окремих функцій веб-сайту, та тестування проводилось послідовно перевіряючи основні сценарії роботи веб-сайту, у процесі тестування перевірялось:

- реєстрація користувачів;
- авторизація користувачів;
- створення запитів;
- перегляд історії запитів;
- взаємодія з ШІ;
- робота адміністративної панелі;
- робота панелі фахівця;
- коректність роботи рольової моделі;
- взаємодія з базою даних.

Основні тестові сценарії веб-сайту (див. табл. 4.1). Також під час тестування було перевірено коректність роботи REST API та обробки HTTP запитів. Для цього виконувались запити на реєстрацію користувача, авторизацію, реєстрацію, створення запитів, роботу з ШІ модулем. Також частину уваги було приділено перевірці роботи автентифікації користувачів. Перевірялась правильність JWT

токенів, доступ сторінок та робота рольової моделі доступу.

Таблиця 4.1 – Основні тестові сценарії веб-сайту

Тестовий сценарій	Очікуваний результат
Реєстрація користувача	Створення новго облікового запису
Авторизація користувача	Успішний вхід до системи
Створення запитів	Додавання запитів до бази даних
Перегляд запитів	Відображення списку запитів
Взаємодія з ШІ	Отримання відповіді від ШІ
Доступ адміністратора	Відкриття адміністративної моделі
Перевірка ролей користувачів	Обмеження доступу до функцій

Результати функціонального тестування представлено у таблиці (див. табл. 4.2)

Таблиця 4.2 – Результати функціонального тестування

Функція	Результат тестування
Реєстрація користувачів	Працює коректно
Авторизація	Працює коректно
Робота із запитами	Працює коректно
REST API	Працює коректно
JWT-автентифікація	Працює коректно
ШІ модуль	Працює коректно
Рольова модель	Працює коректно

Також перевірено коректність роботи на різних пристроях, різних типах при

різних розмірах екрану. Результати цього тестування продемонстрували нам, що веб-сайт коректно працює на різних типах пристроїв, та адаптується до різних роздільних здатностей екрана та забезпечує зручну взаємодію користувача з веб-сайтом з різних пристроїв що додає плюс до зручності користування в будь-якому місці чи то з телефону чи з ноутбука. Результати перевірки роботи на різних пристроях (див. табл 4.3)

Таблиця 4.3 – Результати перевірки роботи на різних пристроях

Тип пристрою	Результат перевірки
Персональний комп'ютер	Працює коректно
Ноутбук	Працює коректно
Планшет	Інтерфейс адаптується коректно
Мобільний пристрій	Інтерфейс адаптується коректно

Виходячи з цього проведено функціональне тестування підтвердило коректність роботи всіх функцій та компонентів, стабільність взаємодії клієнта, сервера, бази даних та ШІ модуля, та відповідність реалізованого функціоналу до вимог.

#### **4.2 Аналіз ефективності та оцінка роботи веб-сайту**

Після функціонального тестування також було зроблено та проведено аналіз ефективності роботи веб-сайту юридичної та психологічної допомоги ветеранам з використанням ШІ. Основним є оцінити швидкість роботи системи, зручності використання, та загалом якість роботи веб-сайту.

Один з головних показників це ефективність веб-сайту при взаємодії клієнтської та серверної частини, за рахунок використання SPA технології на основі Angular, зміст сторінок оновлюється без перезавантаження сторінки, це плюс до продуктивності та досвіду користувача, а також за рахунок серверної частини на

Node.js та Express.js, яка швидко обробляє асинхронні запити та взаємодіє з базою даних, вміст на сторінки отримується дуже швидко тому завантаження всіх даних відбувається майже моментально. Оцінка швидкості основних функцій (див. табл. 4.4)

Таблиця 4.4 – Оцінка швидкості основних функцій

Функція	Результат роботи
Авторизація користувача	Виконується швидко
Завантаження сторінок	Виконується без затримок
Створення запиту	Виконується коректно
Отримання даних із бази даних	Виконується стабільно
Робота ШІ чату	Відповіді формують коректно

Також було проаналізовано зручність використання веб-сайту зі сторони користувача, інтерфейс розроблений з використанням сучасних підходів та забезпечує просто навігацію на веб-сайті між сторінками та функціями. Основним перевагами інтерфейсу є:

- зрозуміла структура сторінок;
- просте навігаційне меню;
- адаптивний дизайн;
- логічне розташування функціональних елементів;
- швидкий доступ до основних функцій веб-сайту.

Також у таблиці (див. табл 4.5) зазначено оцінку зручності використання веб-сайту за основними критеріями оцінки зручності використання веб-сайту юридичної та психологічної допомоги ветеранам з використанням ШІ.

Таблиця 4.5 – Оцінка зручності використання веб-сайту

Перевага	Характеристика
SPA-архітектура	Швидка робота без перезавантаження сторінок
REST API	Зручна взаємодія між компонентами
JWT-автентифікація	Захист доступу до веб-сайту
ШІ модуль	Автоматизована підтримка користувачів
Адаптивний інтерфейс	Підтримка різних типів пристроїв
Модульна структура	Простота підтримки та масштабування

Виходячи з цього за результати аналізу ефективності показали, що розроблений веб-сайт надає стабільну роботу, швидку взаємодію, зручний інтерфейс а також реалізований функціональні можливості відповідають поставленим вимогам та надають ефективну роботу.

### 4.3 Інструкція користувача

Розроблений веб-сайт має простий та зрозумілий інтерфейс, що дає користувачу легко та швидко отримати доступ до всіх основних функцій та можливостей веб-сайту. Для початку роботи користувачу треба пройти процедуру реєстрації та авторизації. Після відкриття веб-сайту користувач потрапляє на головну сторінку де може ознайомитись з призначеннями веб-сайту та перейти до основного функціоналу.

Для створення та реєстрації нового облікового запису необхідно перейти на сторінку реєстрації та заповнити реєстраційну форму, під час реєстрації вказується основна необхідна інформація про користувача. Після реєстрації тепер користувач має можливість увійти до веб-сайту тобто авторизуватись на веб-сайті за допомогою електронної пошти та пароля, після авторизації користувач отримує доступ до основних функцій веб-сайту відповідно до ролі. Послідовність авторизації користувача (див. табл. 4.7)

Таблиця 4.7 – Послідовність авторизації користувача

Крок	Дії користувача
1	Відкрити сторінку авторизації
2	Ввести електронну пошту та пароль
3	Натиснути кнопку Увійти
4	Отримати доступ до веб-сайту

Наступна функція, це функція із запитами користувачів, для отримання юридичної або психологічної допомоги. Для створення запиту користувач переходить до відповідної сторінки де заповнює форму запиту. Після надсилання інформація про запит користувача зберігається у базі даних та доступна для перегляду відповідним фахівцям. Користувач також має змогу переглянути історію своїх запитів та відповідей від фахівців. Створення нового запиту (див. табл. 4.8)

Таблиця 4.8 – Створення нового запиту

Крок	Дія користувача
1	Відкрити сторінку запитів
2	Натиснути кнопку створення запиту
3	Заповнити форму
4	Надіслати запит
5	Отримати підтвердження про успішне створення

Для використання ШІ чат-боту для швидких консультацій, для взаємодії також потрібно перейти на відповідну сторінку веб-сайту з формою спілкування у вигляді чату з ШІ чат-ботом де користувач може вести повідомлення, надіслати його а потім отримати відповідь від ШІ. Робота з ШІ чат-ботом (див.табл. 4.9)

Таблиця 4.9 – Робота з ШІ чат-ботом

Крок	Дія користувача
1	Відкрити сторінку чат-боту
2	Ввести повідомлення
3	Надіслати запит
4	Отримати відповідь від ШІ

Для роботи адміністратора та фахівця також передбачено окремі сторінки які доступні лише адміністраторам чи фахівцям. Фахівці можуть працювати зі зверненнями змінюючи їх статуси, надсилаючи відповіді користувачам, адміністратор може переглядати користувачів веб-сайту та робити відповідну модерацію. Основні можливості ролей користувача (див. табл. 4.10)

Таблиця 4.10 – Основні можливості ролей користувача

Роль	Доступні можливості
Користувач	Створення запитів, перегляд історії, ШІ чат-бот
Фахівець	Перегляд та обробка запитів
Адміністратор	Керування користувачами та веб-сайтом

Виходячи з усього вище переліченого веб-сайт забезпечує простий та зрозумілий процес роботи з ним для кожної категорії та ролі користувачів, реалізований інтерфейс дає змогу швидко та легко отримати доступ до всіх функцій веб-сайту.

#### **Висновок до розділу 4**

У четвертому розділі було проведено тестування, оцінка ефективності веб-сайту юридичної та психологічної допомоги ветеранам з використанням ШІ а також написано інструкцію користувача до основних функціональній можливостей веб-сайту.

Результати всіх аналізів тестування та оцінки ефективності підтвердили працездатність, коректність, швидкість, зручність використання роботи веб-сайту відповідно до поставлених вимог, тому розроблений веб-сайт є працездатним зручним у використанні, та готовим до подальшого удосконалення та розвитку.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено веб-сайт юридичної та психологічної допомоги ветеранам з використанням ШІ, який забезпечує зручну взаємодію зі фахівцями та підтримує можливість отримання первинних консультацій за допомогою модуля штучного інтелекту.

Під час виконання було проведено аналіз предметної області, досліджено існуючі інформаційні ресурси і сервіси, призначенні для підтримки ветеранів. На основі проведеного аналізу та дослідження було визначено основні недоліки існуючих рішень та сформовано вимоги до розроблюваного веб-сайту.

У роботі обгрунтовано вибір технологій розробки та архітектурних рішень, для реалізації клієнтської частини використано фреймворк Angular, серверну частину розроблено на основі Node.js та Express.js, а для зберігання даних користувачів використано системи управління базою даних MySQL. Для забезпечення доступу до функціональних можливостей та забезпечення безпеки було реалізовано JWT-автентифікацію та рольову модель користувачів.

У процесі виконання роботи було спроектовано архітектуру веб-сайту, структуру бази даних та механізми взаємодії між клієнтом, сервером, базою даних та штучним інтелектом. Реалізовано основні функціональні можливості веб-сайту, зокрема реєстрацію, авторизацію користувачів, створення та обробку запитів, особистий кабінет користувача, панель фахівця, адміністративну панель та ШІ чат-бот для автоматизованої підтримки користувачів.

Особливість розробленого веб-сайту є інтеграція модуля штучного інтелекту на основі OpenAI ChatGPT API, який дає змогу надавати користувачу оперативні відповіді на типові запити та здійснювати первинну інформаційну підтримку.

Проведено функціональне тестування, аналіз ефективності та зручності використання підтвердило коректність роботи веб-сайту, стабільність взаємодії між клієнтом, сервером, базою даних та штучним інтелектом, правильність роботи, зручність у використанні та відповідність до вимог.

Отримані результати свідчать про досягнення поставленої цілі кваліфікаційної роботи та виконання визначених завдань. Розроблений веб-сайт може бути використаний як основа для створення повноцінної інформаційної платформи підтримки ветеранів. Подальший розвиток веб-сайту може передбачати розширення функціональних можливостей ШІ модуля, інтеграцію з державними сервісами, впровадження системи сповіщень та розширення механізмів комунікації між користувачами та фахівцями.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Міністерство у справах ветеранів України. Офіційний вебсайт. URL: <https://mva.gov.ua> (дата звернення: 05.05.2026).
2. e-Ветеран. Електронні сервіси для ветеранів. URL: <https://eveteran.gov.ua> (дата звернення: 05.05.2026).
3. Freeman A. *Pro Angular*. – Berkeley : Apress, 2020. – 1030 p.
4. Dayley B. *Node.js, MongoDB and Angular Web Development*. – Indianapolis : Sams Publishing, 2018. – 432 p.
5. Tilkov S., Vinoski S. Node.js: Using JavaScript to Build High-Performance Network Programs // IEEE Internet Computing. – 2010. – Vol. 14, No. 6. – P. 80–83.
6. Angular Documentation. – URL: <https://angular.io> (дата звернення: 05.05.2026)
7. Node.js Documentation. – URL: <https://nodejs.org> (дата звернення: 05.05.2026).
8. Express.js Documentation. – URL: <https://expressjs.com> (дата звернення: 05.05.2026).
9. Microsoft. REST API Design Best Practices. – URL: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design> (дата звернення: 05.05.2026).
10. W3C. Web Standards Overview. – URL: <https://www.w3.org/standards/> (дата звернення: 05.05.2026).
11. PostgreSQL Documentation. – URL: <https://www.postgresql.org> (дата звернення: 05.05.2026). Mozilla Developer Network. Web Security Guidelines. – URL: <https://developer.mozilla.org/en-US/docs/Web/Security> (дата звернення: 05.05.2026).
12. MySQL Documentation. – URL: <https://dev.mysql.com> (дата звернення: 05.05.2026).

13. RiseUp Labs. Node.js Web Development Guide. – URL: <https://riseuplabs.com/nodejs-web-development-ultimate-guide/> (дата звернення: 05.05.2026).
14. Acte. Angular and Node.js Full Stack Development. – URL: <https://www.acte.in/what-is-angular-node> (дата звернення: 05.05.2026).
15. ResearchGate. Modern Web Technology – Frameworks. – URL: [https://www.researchgate.net/publication/390758277\\_Modern\\_web\\_technology\\_frameworks](https://www.researchgate.net/publication/390758277_Modern_web_technology_frameworks) (дата звернення: 05.05.2026).
16. ResearchGate. Scalable Backend Systems using Node.js. – URL: [https://www.researchgate.net/publication/404277283\\_Nodejs\\_backend\\_systems](https://www.researchgate.net/publication/404277283_Nodejs_backend_systems) (дата звернення: 05.05.2026).
17. IBM. Natural Language Processing. – URL: <https://www.ibm.com/topics/natural-language-processing> (дата звернення: 05.05.2026).
18. OpenAI. API Quickstart. – URL: <https://platform.openai.com/docs/quickstart> (дата звернення: 05.05.2026).
19. Google. Machine Learning Crash Course. – URL: <https://developers.google.com/machine-learning/crash-course> (дата звернення: 05.05.2026).
20. Waheed N. et al. *An Empirical Assessment of Security and Privacy Risks of Web-based Chatbots* // arXiv. – 2022. – URL: <https://arxiv.org/abs/2205.08252>
21. Khoury R. et al. *How Secure is Code Generated by ChatGPT?* // arXiv. – 2023. – URL: <https://arxiv.org/abs/2304.09655>
22. Brito T. et al. *Study of JavaScript Static Analysis Tools for Vulnerability Detection* // arXiv. – 2023. – URL: <https://arxiv.org/abs/2301.05097>.
23. CETIC. Architecture for a Chatbot. – URL: <https://www.cetic.be/Architecture-for-a-chatbot> (дата звернення: 05.05.2026).
24. OWASP Foundation. OWASP Top 10: The Ten Most Critical Web Application Security Risks. – URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 05.05.

Mozilla Developer Network. Web Security Guidelines. – URL: <https://developer.mozilla.org/en-US/docs/Web/Security> (дата звернення: 05.05.2026).

25. Microsoft. REST API Design Best Practices. – URL: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design> (дата звернення: 05.05.2026).

26. OWASP Foundation. SQL Injection. – URL: [https://owasp.org/www-community/attacks/SQL\\_Injection](https://owasp.org/www-community/attacks/SQL_Injection) (дата звернення: 05.05.2026).

27. OWASP Foundation. Cross-Site Scripting (XSS). – URL: <https://owasp.org/www-community/attacks/xss/> (дата звернення: 05.05.2026).

28. OWASP Foundation. Cross-Site Request Forgery (CSRF). – URL: <https://owasp.org/www-community/attacks/csrf> (дата звернення: 05.05.2026).

29. Mozilla Developer Network. HTTP Overview. – URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview> (дата звернення: 05.05.2026).

30. Mozilla Developer Network. JSON Guide. – URL: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON> (дата звернення: 05.05.2026).

31. Cloudflare. What is HTTPS? – URL: <https://www.cloudflare.com/learning/ssl/what-is-https/> (дата звернення: 05.05.2026).

32. JWT.io. Introduction to JSON Web Tokens. – URL: <https://jwt.io/introduction> (дата звернення: 05.05.2026).

---

**ДОДАТОК А****Лістинг програмної реалізації основних компонентів вебсайту**

“C:\veterans-support-system\src\app\app.routes.ts”

```
export const routes: Routes = [
  { path: '', component: Home },
  { path: 'login', component: Login },
  { path: 'register', component: Register },
  {
    path: 'profile',
    component: Profile,
    canActivate: [AuthGuard],
  },
  {
    path: 'requests',
    component: Requests,
    canActivate: [AuthGuard],
  },
  {
    path: 'requests/:id',
    component: RequestDetails,
    canActivate: [AuthGuard],
  },
  {
    path: 'chat',
    component: Chat,
    canActivate: [AuthGuard],
  },
  {
    path: 'specialist',
    component: SpecialistDashboard,
    canActivate: [AuthGuard, RoleGuard],
    data: { roles: [UserRole.Specialist, UserRole.Administrator] },
  },
  {
    path: 'admin',
    component: Admin,
    canActivate: [AuthGuard, RoleGuard],
    data: { roles: [UserRole.Administrator] },
  },
  { path: '**', redirectTo: '' },
];
```

“C:\veterans-support-system\src\app\services\auth.service.ts”

```
export class AuthService {
  private readonly authState = new BehaviorSubject<boolean>(this.hasToken());

  constructor(
    private readonly http: HttpClient,
    private readonly mockDataService: MockDataService,
```

```
) {}

login(payload: LoginPayload): Observable<AuthResponse> {
  return this.http
    .post<AuthResponse>(API_ENDPOINTS.auth.login, payload)
    .pipe(
      catchError(() => this.mockLogin(payload)),
      tap((response) => this.persistSession(response)),
    );
}

register(payload: RegisterPayload): Observable<AuthResponse> {
  return this.http
    .post<AuthResponse>(API_ENDPOINTS.auth.register, payload)
    .pipe(
      catchError(() => this.mockRegister(payload)),
      tap((response) => this.persistSession(response)),
    );
}

logout(): void {
  localStorage.removeItem(TOKEN_STORAGE_KEY);
  localStorage.removeItem(USER_STORAGE_KEY);
  this.authState.next(false);
}

isAuthenticated(): boolean {
  return this.authState.value;
}

authChanges(): Observable<boolean> {
  return this.authState.asObservable();
}

getToken(): string | null {
  return localStorage.getItem(TOKEN_STORAGE_KEY);
}

getCurrentUser(): User | null {
  const serialized = localStorage.getItem(USER_STORAGE_KEY);
  if (!serialized) {
    return null;
  }

  const user = JSON.parse(serialized) as User;
  return {
    ...user,
    role: this.normalizeRole(user.role as string),
  };
}

getCurrentRole(): UserRole {
  return this.getCurrentUser()?.role ?? UserRole.User;
}

hasRole(role: UserRole): boolean {
```

```

return this.getCurrentRole() === role;
}

hasAnyRole(roles: UserRole[]): boolean {
  return roles.includes(this.getCurrentRole());
}

private persistSession(response: AuthResponse): void {
  const normalized: AuthResponse = {
    ...response,
    user: {
      ...response.user,
      role: this.normalizeRole(response.user.role as string),
    },
  };

  localStorage.setItem(TOKEN_STORAGE_KEY, normalized.token);
  localStorage.setItem(USER_STORAGE_KEY, JSON.stringify(normalized.user));
  this.authState.next(true);
}

private hasToken(): boolean {
  return !!localStorage.getItem(TOKEN_STORAGE_KEY);
}

private mockLogin(payload: LoginPayload): Observable<AuthResponse> {
  return this.mockDataService.getUsers().pipe(
    map((users) => {
      const fallbackUsers = users.length > 0 ? users : MOCK_USERS;
      const matched =
        fallbackUsers.find((entry) => entry.email.toLowerCase() ===
payload.email.toLowerCase()) ??
        fallbackUsers[0];

      return {
        token: `demo-token-${matched.id}`,
        user: matched,
      } as AuthResponse;
    })
  );
}

private mockRegister(payload: RegisterPayload): Observable<AuthResponse> {
  const created: User = {
    id: Date.now(),
    fullName: payload.fullName,
    email: payload.email,
    role: UserRole.User,
    createdAt: new Date().toISOString(),
  };

  return of({
    token: `demo-token-${created.id}`,
    user: created,
  });
}

```

```
private normalizeRole(role: string): UserRole {  
    if (role === UserRole.Administrator || role === 'admin') {  
        return UserRole.Administrator;  
    }  
    if (role === UserRole.Specialist) {  
        return UserRole.Specialist;  
    }  
  
    return UserRole.User;  
}  
}
```

Наведені лістинги демонструють основний код компонентів вебсайту такі як маршрутизація, авторизація та реєстрація користувачів.