

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

_____Юрій КОНДРАТЕНКО

«____»_____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

**ВЕБЗАСТОСУНОК ДЛЯ СЛУЖБИ ПСИХОЛОГІЧНОЇ
ПІДТРИМКИ ОСІБ**

Спеціальність 122 Комп'ютерні науки

Освітня програма «Комп'ютерні науки»

Здобувач

_____Надія КУДРЯ

«____»_____ 2025 р.

Керівник канд. фіз.-мат. наук, доцент

_____Інесса КУЛАКОВСЬКА

«____»_____ 2025 р.

Миколаїв – 2025

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

_____ Юрій КОНДРАТЕНКО

« ____ » _____ 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Кудрі Надії Сергіївни

((прізвище, ім'я, по батькові здобувача))

1. Тема кваліфікаційної роботи: «Вебзастосунок для служби психологічної підтримки осіб».

Керівник роботи: Кулаковська Інесса Василівна, доцент кафедри інтелектуальних інформаційних систем, кандидат фіз.-мат. наук.

Затверджена наказом ЧНУ ім. Петра Могили від «18» листопада 2024 р. № 321.

2. Строк представлення кваліфікаційної роботи « ____ » _____ 2025 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: розроблений адаптивний вебзастосунок для надання психологічної підтримки на основі аналізу

існуючих вебсервісів психологічної підтримки, технічних вимог до функціоналу, конфіденційності даних і обраних технологій (React, Node, MongoDB).

4. Перелік питань, що підлягають розробці: аналіз потреб користувачів у сфері онлайн-психологічної підтримки; огляд існуючих вебсервісів та визначення їх сильних і слабких сторін; розробка структури бази даних; застосування заходів щодо забезпечення конфіденційності та безпеки даних; створення API для обміну даними між клієнтською та серверною частинами.

5. Перелік графічних матеріалів: таблиці, рисунки, презентація.

Керівник роботи

(Особистий підпис)

Інесса КУЛАКОВСЬКА
(Власне ім'я ПРІЗВИЩЕ)

Здобувач

(Особистий підпис)

Надія КУДРЯ
(Власне ім'я ПРІЗВИЩЕ)

Дата видачі завдання «18» грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

кваліфікаційної роботи

Тема: Вебзастосунок для служби психологічної підтримки осіб

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	18.12.2024	20.12.2024	
2	Аналіз предметної області та постановка задачі	21.12.2024	30.01.2025	
3	Огляд літературних джерел за темою кваліфікаційної роботи, зокрема огляд публікацій та аналогічних систем щодо вебзастосунків психологічної підтримки	31.01.2025	01.03.2025	
4	Огляд існуючих архітектур вебзастосунків та технологій для реалізації сервісів психологічної підтримки	02.03.2025	01.04.2025	
5	Реалізація обраних технологій із аналізом отриманих результатів у вебзастосунку для психологічної підтримки осіб	02.04.2025	15.05.2025	
6	Перший попередній захист КР на засіданні комісії кафедри	16.05.2025	16.05.2025	
7	Корегування роботи за результатами попереднього захисту	29.05.2025	29.05.2025	
8	Другий попередній захист КР на засіданні комісії кафедри	30.05.2025	30.05.2025	
9	Доробка та остаточне оформлення КР	31.05.2025	10.06.2025	
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	13.06.2025	13.06.2025	

Керівник роботи

(Особистий підпис)

Інеса КУЛКОВСЬКА

(Власне ім'я ПРІЗВИЩЕ)

Здобувач

(Особистий підпис)

Надія КУДРЯ

(Власне ім'я ПРІЗВИЩЕ)

Дата складання календарного плану

«29» січня 2025 р.

АНОТАЦІЯ

до кваліфікаційної роботи
здобувачки групи 401 ЧНУ ім. Петра Могили

Кудрі Надії Сергіївни

на тему: «**ВЕБЗАСТОСУНОК ДЛЯ СЛУЖБИ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ
ОСІБ**»

Актуальність даного дослідження полягає в необхідності розвитку цифрових сервісів у сфері охорони психічного здоров'я за високим соціальним запитом на доступну психологічну підтримку в період війни.

Вебзастосунок забезпечує можливості для онлайн-консультацій, анонімних звернень, спілкування з фахівцями в режимі реального часу, а також доступ до інформаційних матеріалів і порад щодо психологічного самоцілення.

Об'єктом роботи є система надання онлайн психологічної підтримки особам, які цього потребують.

Предметом роботи є вебтехнології та засоби реалізації адаптивного застосунку для психологічної підтримки.

Метою роботи є підвищення доступності психологічної допомоги шляхом створення функціонального вебзастосунку, який забезпечує оперативний зв'язок між користувачами й психологами, автоматизацію запису на консультацію та можливість анонімного звернення.

Пояснювальна записка бакалаврської роботи складається зі вступу, чотирьох розділів, висновків і 3 додатків. У першому розділі розглянуто предметну область, описано потребу в створенні цифрового продукту для психологічної підтримки та проаналізовано існуючі аналоги. У другому розділі проаналізовано та обґрунтовано вибір технологій: React.js для інтерфейсу користувача, Node.js та Express для сервера, MongoDB для зберігання даних, а також JWT для авторизації та Socket.io для реалізації чату в реальному часі. У третьому розділі представлено проєктування системи: описано структуру бази даних, компоненти інтерфейсу, сценарії взаємодії

користувачів і психологів, спроектовано REST API. У четвертому розділі здійснено програмну реалізацію застосунку, наведено приклади фрагментів коду, описано логіку функціональних модулів.

Бакалаврська кваліфікаційна робота містить 71 сторінок, 36 рисунків, 10 таблиць, 30 джерел посилення та 5 додатки.

Ключові слова: психологічна підтримка, вебзастосунок, React.js, Node.js, MongoDB, REST API, Socket.io, JWT.

ABSTRACT

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

Kudria Nadiia

"WEB APPLICATION FOR THE PSYCHOLOGICAL SUPPORT SERVICE OF INDIVIDUALS"

A relevance of this study lies in the need to develop digital services in the field of mental health care due to the high social demand for accessible psychological support during the war. The web application provides opportunities for online consultations, anonymous appeals, communication with specialists in real time, as well as access to information materials and advice on psychological self-healing. An object of the work is a system for providing online psychological support to individuals in need.

A object of the work is a system for providing online psychological support to individuals in need.

A subject of the work is web technologies and means of implementing an interactive application for psychological support.

A purpose of the work is to increase the accessibility of psychological help by creating a functional web application that provides operational communication between users and psychologists, automation of appointment booking and the possibility of anonymous appeal.

The explanatory note of the bachelor's thesis consists of an introduction, four sections, conclusions and 3 appendices. The first section considers the subject area, describes the need to create a digital product for psychological support and analyzes existing analogues. The second section analyzes and justifies the choice of technologies: React.js for the user interface, Node.js and Express for the server, MongoDB for data storage, as well as JWT for authorization and Socket.io for implementing real-time chat. The third section presents the

system design: the database structure, interface components, user and psychologist interaction scenarios are described, and the REST API is designed. The fourth section provides software implementation of the application, provides examples of code fragments, and describes the logic of functional modules.

The qualification work contains 71 pages, 36 figures, 10 tables, 30 reference sources and 5 applications.

Keywords: psychological support, web application, React.js, Node.js, MongoDB, REST API, Socket.io, JWT.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	4
ВСТУП.....	5
1 АНАЛІЗ СУЧАСНОГО СТАНУ ПОСТАВЛЕНОЇ ЗАДАЧІ	6
1.1 Опис предметної сфери	6
1.2 Аналіз існуючих рішень та огляд наукових публікацій	7
1.3 Постановка задачі.....	15
Висновки до розділу 1	17
2 МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	19
2.1 Обґрунтування вибору методів для розв’язання поставлених задач.....	19
2.2 Технології розробки системи.....	23
Висновки до розділу 2	24
3 РОЗРОБКА ВЕБЗАСТОСУНКУ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ	26
3.1 Аналіз вхідних даних та структура вебзастосунку.....	26
3.2 Проєктування та моделювання вебзастосунку служби психологічної підтримки	28
Висновки до розділу 3	40
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА ДОКУМЕНТАЦІЇ	41
4.1 Опис програмної реалізації	41
4.2 Структура проєкту	47

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

4.3 Тестування ПЗ	61
Висновки до розділу 4.....	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	69
ДОДАТОК А Діаграма варіантів використання для акторів.....	72
ДОДАТОК Б Карта вебзастосунку для клієнта	73
ДОДАТОК В Структура серверної частини проєкту	74
ДОДАТОК Г Маршрути для обробки запитів, пов'язаних з користувачами.....	77
ДОДАТОК Д Реалізація інтеграції LiqPay для онлайн-оплат консультацій	82

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API	–	Application Programming Interface
HTML	–	HyperText Markup Language
CSS	–	Cascading Style Sheets
JS	–	JavaScript
UML	–	Unified Modeling Language
REST	–	REpresentational State Transfer

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

ВСТУП

У теперішніх умовах повномасштабної війни в Україні значна кількість населення стикається з високим рівнем психологічного навантаження, тривогою, депресією, посттравматичним стресовим розладом та іншими різновидами психоемоційної дестабілізації. Відсутність оперативного доступу до фахової психологічної підтримки ускладнює процес здолання цих станів, особливо у віддалених або тимчасово окупованих областях.

Одним із ефективних способів підтримки громадян є створення вебзастосунку, який забезпечує зручне спілкування з фахівцями, дає змогу отримати консультацію онлайн, пройти первинне тестування стану, а також ознайомитися з корисними матеріалами. Вебзастосунок дає змогу кожному користувачеві звернутися анонімно або ідентифіковано, отримати своєчасну допомогу без потреби відвідування фізичного кабінету спеціаліста.

Особливу увагу слід приділити безпеці даних користувачів: вебзастосунок має відповідати сучасним стандартам захисту персональної інформації, підтримувати шифрування повідомлень і зберігати конфіденційність усіх звернень.

Таким чином, створення такого вебзастосунку є не лише актуальним кроком у напрямку подолання кризи психічного здоров'я серед населення, а й важливим елементом цифрової трансформації сфери психологічної допомоги в умовах війни.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1 Опис предметної сфери

Психологічне здоров'я — одна з ключових умов повноцінного функціонування особистості, її здатності до адаптації, самореалізації, а також ефективної взаємодії в соціумі. За визначенням Всесвітньої організації охорони здоров'я (ВООЗ), психологічне здоров'я — це стан психічного благополуччя, який дозволяє людям справлятися зі стресами життя, реалізовувати свої здібності, добре навчатися та працювати, а також робити внесок у розвиток своєї громади [1].

В умовах повномасштабної війни в Україні спостерігається різке зростання кількості осіб, які відчують психологічні труднощі. За даними Міністерства охорони здоров'я України, близько 60% громадян мають депресію та інші психічні розлади внаслідок війни. Найвразливішими категоріями є внутрішньо переміщені особи, військові, їхні родини, діти та підлітки.

У зв'язку з цим зростає потреба в цифрових інструментах, які б забезпечували швидкий, безпечний та зручний доступ до психологічної допомоги. Онлайн-сервіси надають користувачам можливість звертатися до фахівців анонімно, без обмежень у просторі та часі. Це дозволяє мінімізувати бар'єри до отримання допомоги, зокрема серед осіб, які не готові звертатися до психолога особисто.

На міжнародному рівні ефективність онлайн-психотерапії підтверджують такі сервіси, як BetterHelp, Talkspace тощо. В Україні аналогічну функцію виконують платформи «Розкажи мені», «Ти як» тощо.

Втім, значна частина існуючих рішень не забезпечує повноцінну ефективність, не має гнучкого механізму зворотного зв'язку або не підтримує запис на індивідуальні сесії. Це створює необхідність розробки нового програмного рішення з підтримкою реального часу, розширеними можливостями спілкування, авторизацією користувачів

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

та зручним інтерфейсом.

Таким чином, предметна сфера цієї роботи охоплює питання психологічної допомоги у кризових умовах, функціональні потреби онлайн-комунікації між спеціалістами та користувачами, а також аналіз технологій, що можуть бути застосовані для створення ефективного вебзастосунку.

1.2 Аналіз існуючих рішень та огляд наукових публікацій

У зв'язку з ростом психологічної напруги в суспільстві, особливо в умовах воєнного стану, зросла актуальність розробки та використання веборієнтованих інструментів для надання дистанційної психологічної підтримки. Проаналізуємо низку відомих вебзастосунків, які мають на меті полегшити стан користувачів шляхом надання психологічних сервісів онлайн.

1. **BetterHelp.**

Однією з найвідоміших міжнародних платформ є **BetterHelp** [2] — це комерційна онлайн-платформа, яка пропонує користувачам спілкування із ліцензованими психотерапевтами.

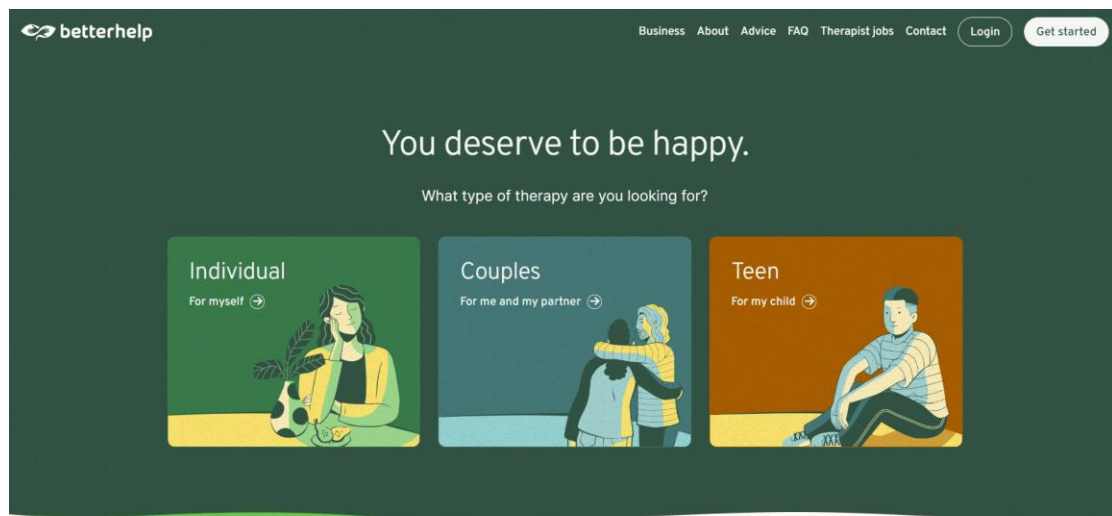


Рисунок 1.1 – Головна сторінка вебсайту BetterHelp

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

Даний сервір пропонує зручні формати взаємодії: текстовий чат, відео та аудіо сеанси. Користувач проходить опитування, після чого система підбирає йому спеціаліста.

Перевагами є велика база фахівців, індивідуальний підхід та можливість гнучкого графіку консультацій. Проте сервіс є дорогим, особливо для країн з низьким рівнем доходів.

Крім того, відсутність підтримки української мови та адаптованих протоколів для роботи з потерпілими від війни значно знижує ефективність цього інструменту в українському контексті.

2. Talkspace.

На відміну від BetterHelp, Talkspace [3] пропонує структуровані терапевтичні програми, створені відповідно до сучасних стандартів когнітивно-поведінкової терапії. Крім психотерапевтів, Talkspace має ліцензованих психіатрів, які можуть призначати медикаментозне лікування.

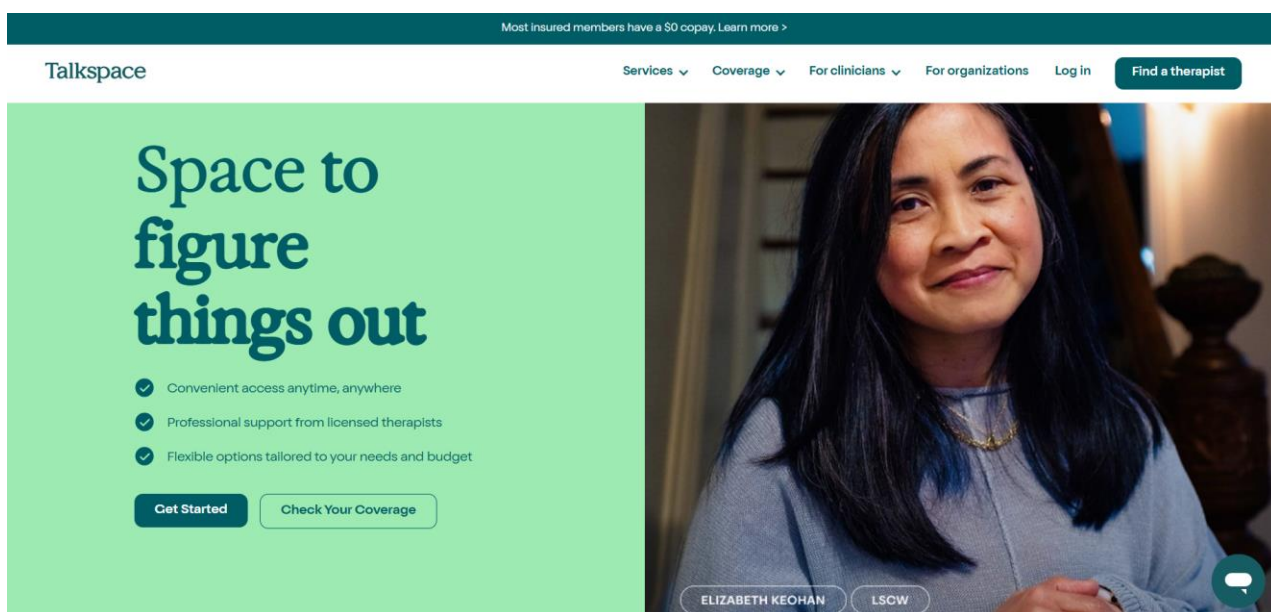


Рисунок 1.2 – Головна сторінка вебсайту Talkspace

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

Доступ до консультацій відбувається через спеціальний додаток, що дозволяє вести листування або брати участь у сесіях. Незважаючи на високий рівень професіоналізму, платформа також страждає від недостатньої мовної адаптації, відсутності українських фахівців і надмірної вартості.

3. «Ти як?».

На відміну від комерційних рішень, українські ініціативи орієнтовані переважно на безкоштовну допомогу населенню, яке постраждало внаслідок війни. Прикладом такої ініціативи є платформа «Ти як?» [4], що пропонує добірку технік саморегуляції, дихальних вправ, практик зниження тривожності, стабілізації психоемоційного стану.

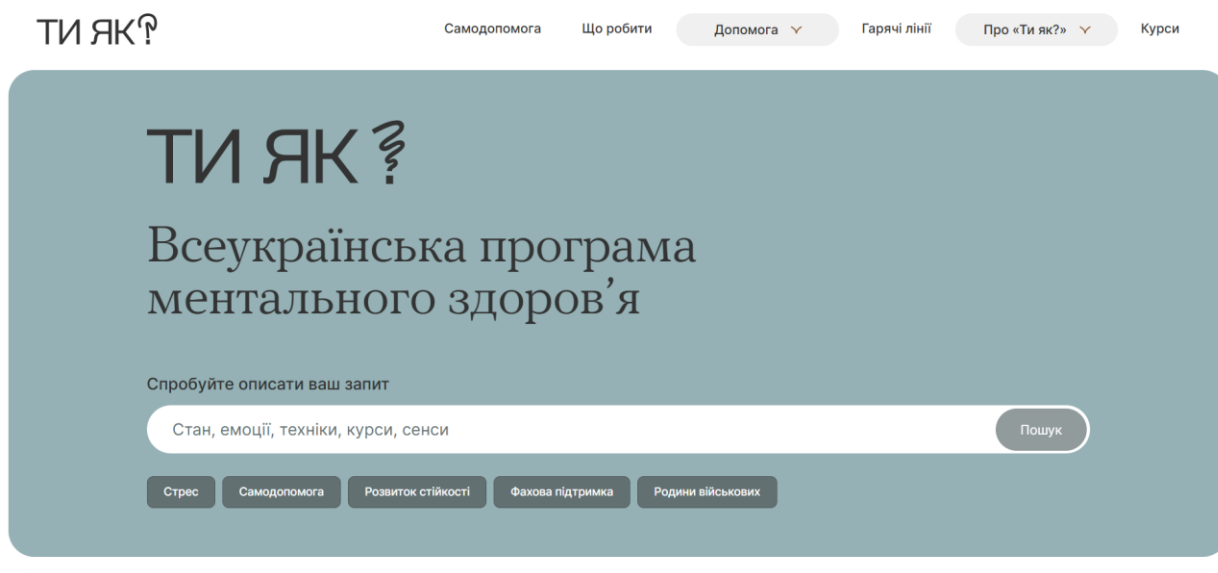


Рисунок 1.3 – Головна сторінка вебсайту «Ти як?»

Усі матеріали україномовні, а доступ до них не потребує реєстрації. Проте сервіс не передбачає прямого контакту з фахівцем. Його розвиток міг би включати інтерактивність — наприклад, запровадження чат-бота чи відеоінструкцій.

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

4. «Розкажи мені».

Ще одним українським ресурсом є платформа «**Розкажи мені**» [5], яка дає можливість користувачам отримати підтримку від реальних психологів. Вона орієнтована на тих, хто зазнав травматичного досвіду, пов'язаного з війною, і надає безкоштовні індивідуальні консультації.

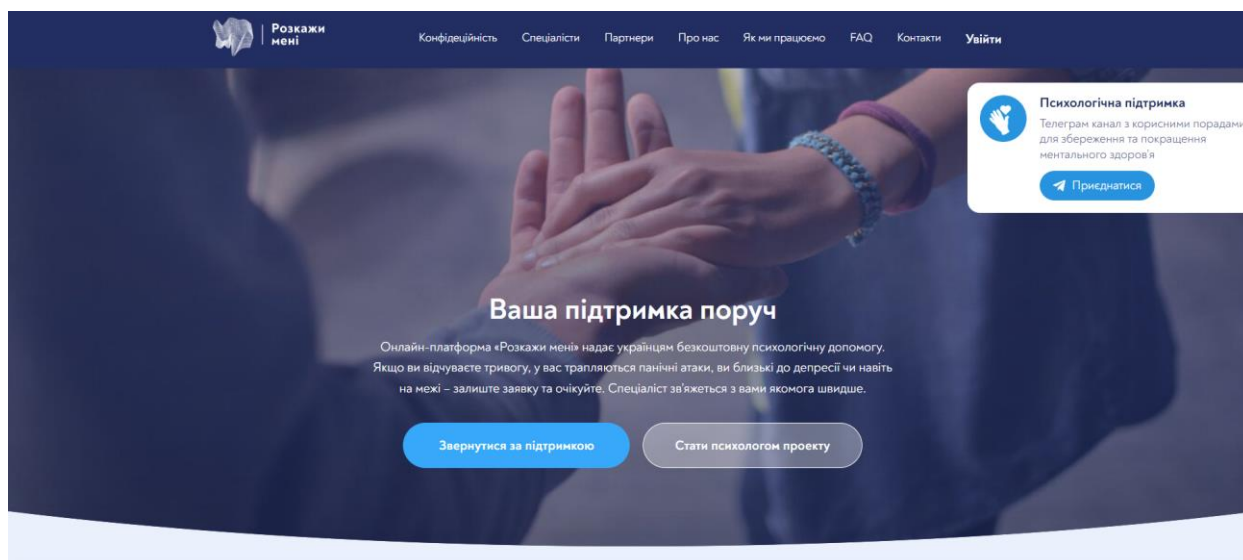


Рисунок 1.4 – Головна сторінка вебсайту «Розкажи мені»

Проте через велику кількість звернень спостерігається дефіцит фахівців, а відсутність мобільного додатку зменшує зручність використання. Розвиток цього сервісу може включати розширення команди спеціалістів, а також створення мобільного інтерфейсу з вбудованими модулями самопомоги.

5. Mindly.

Серед нових українських рішень варто згадати платформу **Mindly** [6], що поєднує цифровий щоденник емоцій, вправи саморегуляції та можливість звернення до психолога. Вона має зручний мобільний додаток, орієнтована на молодь та переселенців, пропонує україномовний інтерфейс та просту реєстрацію.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

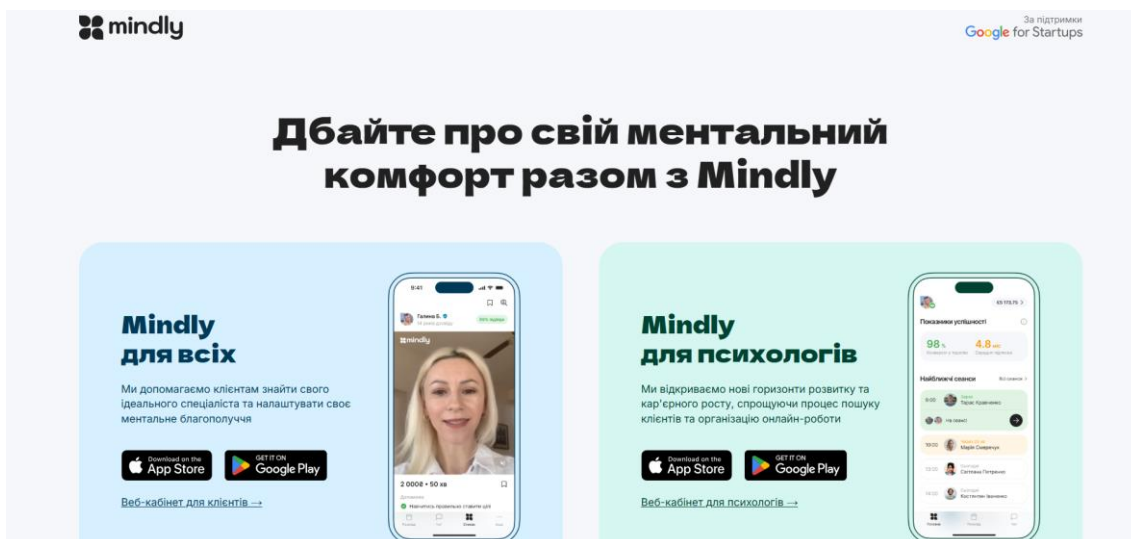


Рисунок 1.5 – Головна сторінка вебсайту Mindly

Mindly дозволяє вести особисті записи про емоційний стан, після чого система пропонує рекомендації — наприклад, дихальну практику, прогулянку або звернення до фахівця.

Хоча платформа ще перебуває в розвитку, її сильними сторонами є адаптація до українських реалій, орієнтація на самодопомогу та цифрову грамотність. Недоліком залишається обмежена база спеціалістів і відсутність анонімного звернення.

Наукові дослідження останніх років також підтверджують високу ефективність цифрових рішень у сфері психологічної допомоги. Нижче наведено результати аналізу кількох публікацій, що є релевантними до теми роботи.

1. Кваліфікаційна робота «Вебзастосунок для психологічної підтримки під час війни». Автор: А. В. Гончаренко, ЧНУ ім. П. Могили, 2023 [7].

Кваліфікаційна робота присвячена розробці вебзастосунку для надання психологічної підтримки під час війни. Основною метою є оптимізація процесу

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

отримання психологічної підтримки шляхом зменшення часу організацій психологічних консультацій у вебзастосунку підтримки під час війни.

У межах роботи було проаналізовано існуючі сервіси психологічної підтримки, визначено потреби цільової аудиторії в умовах воєнного стану та спроектовано архітектуру застосунку з урахуванням сучасних вимог. Проєкт містить в собі інтерактивне самотестування, корисні матеріали, а також можливість надсилання зворотнього зв'язку.

Для створення вебзастосунку були використані такі технології: HTML5, CSS3, JavaScript та React.js для клієнтської частини, Node.js та Express.js для серверної частини, MongoDB для збереження даних, а також Figma для розробки дизайну інтерфейсу. Забезпечено авторизацію за допомогою JWT, а також розгортання на хмарній платформі.

Серед переваг роботи варто відзначити вдалий підхід до аналізу потреб користувачів, застосування сучасних вебтехнологій та створення адаптивного та безпечного продукту, готового до реального використання. Але, застосунок орієнтований переважно на загальний підхід до підтримки й не розраховує різні рівні психологічних розладів. Питання конфіденційності та збереження персональних даних розкрито поверхнево, хоча для такого типу сервісу воно є критично важливим. Також відсутнє тестування проєкту на реальних користувачах, що не дозволяє повноцінно оцінити його зручність і ефективність у практиці.

2. Кваліфікаційна робота «Розробка програмного засобу для щоденної психологічної підтримки користувача мовою Python». Автор: П. С. П'явчук, ДУІКТ, 2024 [8].

У роботі розглядається процес створення програмного застосунку, який забезпечує щоденну психологічну підтримку користувача. Насамперед було проведено аналіз потреб цільової аудиторії та існуючих програмних рішень у сфері

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

самопідтримки. Метою є надання користувачам прості та ефективні інструменти для самостійного відстеження їхнього психологічного стану.

Застосунок включає щоденник настрою, нагадування про релаксаційні вправи, інтерфейс для щоденної взаємодії з користувачем, а також інструменти візуалізації динаміки змін настрою. Технічна реалізація здійснена мовою Python із застосуванням бібліотек tkinter для графічного інтерфейсу, sqlite3 для збереження даних, matplotlib для побудови графіків, а також опціонально smtplib для відправлення нагадувань на електронну пошту. Програма орієнтована на користувачів Windows як настільний додаток.

Перевагами розробки є простота та зручність інтерфейсу, можливість отримання рекомендацій залежно від стану користувача, а також комплексний підхід до підтримки психологічного здоров'я за допомогою щоденних інтерактивних повідомлень і вправ.

Однак робота має певні обмеження. Застосунок орієнтований лише на платформу Windows, що обмежує аудиторію користувачів, які працюють на інших операційних системах або мобільних пристроях. Функціонал застосунку, хоч і достатній для базової підтримки, не передбачає глибшого аналізу психологічних станів або інтеграції з професійними консультаціями, що зменшує можливості для комплексної допомоги. Також відсутня автоматична синхронізація даних між пристроями, що ускладнює використання застосунку в різних середовищах.

3. Кваліфікаційна робота «Розробка веб-платформи для ведення блогу психологічної підтримки». Автор: І. М. Шпирналь, УКД, 2024 [9].

Кваліфікаційна робота присвячена створенню веб-платформи для ведення блогу з психологічної підтримки. Основною метою є надання зручного інструменту для публікації матеріалів на теми психічного здоров'я, самодопомоги, подолання стресу, а також для побудови взаємодії з читачами через коментарі, зворотний зв'язок та

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

підписки. Проєкт орієнтований на широку аудиторію, включаючи як фахівців-психологів, так і людей, які шукають підтримку в складних життєвих обставинах.

В процесі роботи проведено аналіз існуючих блогівих платформ із психоемоційною тематикою, що дозволило виявити їхні сильні та слабкі сторони та сформулювати вимоги до нового рішення з урахуванням доступності, простоти користування та безпеки. Розроблено архітектуру застосунку, що включає основні функціональні модулі: публікації, коментування, підписки та адміністрування контенту. Технічна реалізація виконана з використанням сучасних вебтехнологій — React.js для клієнтської частини, Node.js і Express.js на сервері, MongoDB для збереження даних, а також JWT для авторизації. Інтерфейс розроблено з урахуванням UX/UI принципів, дизайн створений у Figma. Платформа розгорнута на хмарному сервісі Vercel.

Серед переваг розробки варто виділити інтуїтивний та адаптивний інтерфейс, багаторівневу систему авторизації, що підвищує безпеку, а також наявність базових механізмів модерації, які допомагають підтримувати якість контенту та безпеку користувачів.

До недоліків можна віднести обмежену кількість функцій модерації, що може створювати ризики появи небажаного контенту без додаткових інструментів автоматичного контролю. Також платформа не містить розширених аналітичних інструментів для авторів, які могли б допомогти краще зрозуміти аудиторію і підвищити якість матеріалів. Відсутність інтеграції з мобільними застосунками обмежує комфорт користування для мобільних користувачів, що є суттєвим у сучасних умовах.

Аналіз існуючих вебзастосунків та публікацій для надання психологічної підтримки дозволяє виокремити як сильні сторони сучасних рішень, так і низку істотних недоліків.

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

Зважаючи на це, доцільно у рамках проекту розробити вебзастосунок, що враховує потреби українських користувачів у воєнний час. Такий застосунок може включати наступні ключові можливості.

1. Повна україномовна локалізація.
2. Модулі самостійної роботи — цифровий щоденник емоційного стану та тести для самодіагностики.
3. Функція зв'язку з фахівцем, яка дозволяє користувачеві звернутися до впсихолога, з можливістю вибору формату взаємодії (чат, відеоконференція).
4. Мобільна адаптація, що розширить доступність платформи для користувачів, які мають лише смартфон.
5. Анонімний режим використання, який гарантує конфіденційність та зменшує психологічний бар'єр при зверненні по допомогу.

1.3 Постановка задачі

Актуальність теми визначається необхідністю розвитку цифрових сервісів у сфері охорони психічного здоров'я, а також високим соціальним запитом на доступну психологічну підтримку в період кризових ситуацій.

Об'єктом кваліфікаційної роботи є система надання онлайн психологічної підтримки особам, які цього потребують.

Предметом кваліфікаційної роботи є вебтехнології та засоби реалізації інтерактивного застосунку для психологічної підтримки.

Метою кваліфікаційної роботи є підвищення доступності психологічної допомоги шляхом створення функціонального вебзастосунку, який забезпечує оперативний зв'язок між користувачами й психологами, автоматизацію запису на консультацію та можливість анонімного звернення.

Для досягнення мети було поставлено такі основні завдання:

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

- проаналізувати існуючі сервіси психологічної підтримки;
- обґрунтувати вибір технологій для реалізації вебзастосунку;
- спроектувати архітектуру системи;
- реалізувати програмну частину клієнта та сервера;
- протестувати функціональні можливості системи;
- оцінити ефективність і перспективи подальшого розвитку системи.

У межах розробки вебзастосунку визначено набір основних модулів, кожен із яких виконує певну функціональну роль і забезпечує користувача необхідними сервісами для взаємодії з платформою.

Початковим етапом є реалізація модуля реєстрації та авторизації, що дозволяє створювати облікові записи для різних типів користувачів: клієнтів, психологів та адміністратора. Авторизовані користувачі отримують доступ до персоналізованих функцій відповідно до своєї ролі в системі.

Після входу користувачі потрапляють до особистого кабінету. Для клієнтів у кабінеті доступна інформація про власні звернення, майбутні консультації та особисті дані, які можна редагувати. Для психологів особистий кабінет містить функціонал управління розкладом, перегляду запитів, спілкування з клієнтами через чат.

Одним із важливих функціональних блоків є модуль тестування психічного стану. Він надає користувачам змогу пройти психологічний тест, отримати автоматично сформовані результати та рекомендації на їх основі.

Для комунікації між клієнтами та фахівцями реалізовано онлайн-чат. Він забезпечує обмін повідомленнями в реальному часі та збереження історії діалогу. Така форма зв'язку є зручною для коротких консультацій або уточнень.

Окремий модуль відповідає за запис на консультацію. Він дозволяє користувачам обирати доступного спеціаліста, переглядати його графік та бронювати

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

зручний час. Для психологів цей функціонал відображає актуальний розклад і дозволяє керувати своїм часом.

З метою розширення доступності платформи передбачено анонімну форму звернення. Вона дає можливість користувачам, які не бажають реєструватися, відправити повідомлення про допомогу. Після чого користувача відправляють в анонімний чат, куди також підключається психолог. Листування відбувається в реальному часі.

Інформаційна підтримка реалізована у вигляді блогу, де публікуються корисні матеріали, поради від фахівців, новини у сфері психології. Це дозволяє користувачам отримувати базові знання та підтримку навіть без прямого звернення до спеціаліста.

Адміністративна частина системи — це панель адміністратора, через яку здійснюється керування платформою: контроль за активністю користувачів та психологів, а також технічне обслуговування.

Розгортання модулів здійснюється поетапно, починаючи з базового функціоналу (реєстрація, кабінети користувачів), поступово переходячи до інструментів взаємодії (чат, консультації), а далі — до розширених функцій, таких як блог, тестування та адміністрування.

Висновки до розділу 1

У результаті проведеного аналізу встановлено, що в умовах зростаючої психологічної напруги, викликаній повномасштабною війною в Україні, питання доступності, ефективності та безпечності психологічної підтримки набуває особливої актуальності. Існуючі онлайн-сервіси, як міжнародні (BetterHelp, Talkspace, Woebot), так і українські («Ти як?», «Розкажи мені», Mindly), частково покривають потреби користувачів, однак мають низку обмежень — зокрема, мовний бар'єр, відсутність анонімності, недостатню інтерактивність або слабку підтримку в реальному часі.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Результати огляду підтверджують доцільність створення нового вебзастосунку, орієнтованого на українських користувачів, з урахуванням специфіки воєнного часу. Такий застосунок має забезпечувати анонімне спілкування зі спеціалістами, інтерактивну взаємодію, підтримку в реальному часі, адаптацію до мобільних пристроїв, а також україномовний інтерфейс.

2 МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Обґрунтування вибору методів для розв’язання поставлених задач

Для вирішення поставленої задачі було використано кілька сучасних методів, спрямованих на забезпечення безпеки, ефективної авторизації користувачів та зручного візуального представлення даних.

Одним із ключових компонентів, реалізованих у системі, став метод шифрування паролів за допомогою алгоритму bcrypt. Цей алгоритм спеціально розроблений для захисту паролів і є стійким до атак перебору (brute-force) та атак за словником. Його особливістю є використання "солі" — випадкового набору символів, який додається до кожного пароля перед хешуванням. Це забезпечує унікальність хешів навіть для однакових паролів і запобігає створенню попередньо обчислених таблиць (rainbow tables).

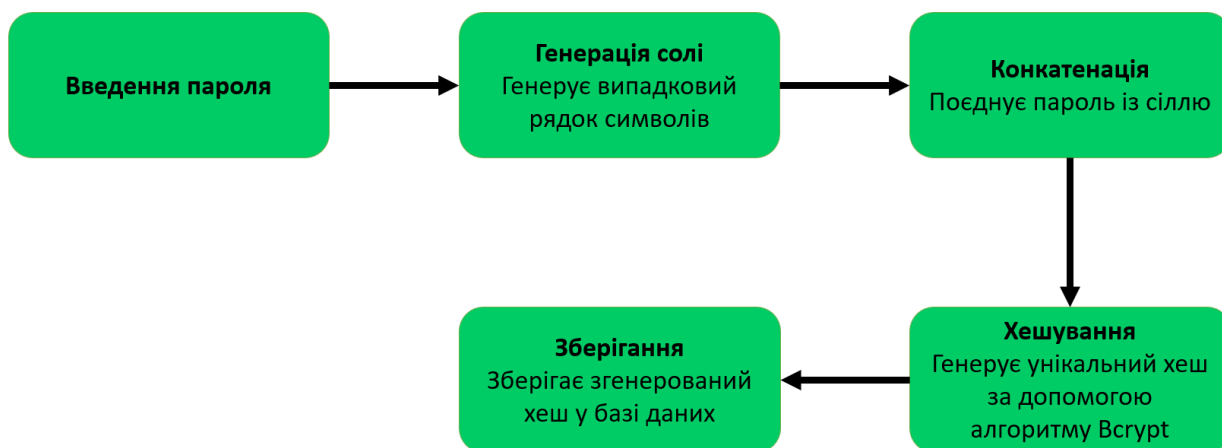


Рисунок 2.1 – Процес хешування за допомогою алгоритму bcrypt

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Крім того, bcrypt дозволяє гнучко налаштовувати складність обчислень, що ускладнює масову компрометацію даних у разі злому бази. У результаті, навіть якщо сторонній користувач отримає доступ до бази даних, він не зможе отримати справжні паролі, оскільки алгоритм не передбачає можливості зворотного дешифрування.

Наступним важливим методом, використаним у роботі, є використання токенів авторизації, зокрема — JWT) Цей підхід забезпечує безпечну та ефективну модель авторизації, у якій після проходження аутентифікації користувач отримує спеціальний цифровий токен. Токен містить задовану інформацію про користувача, його роль у системі, час дії токена та інші метадані.

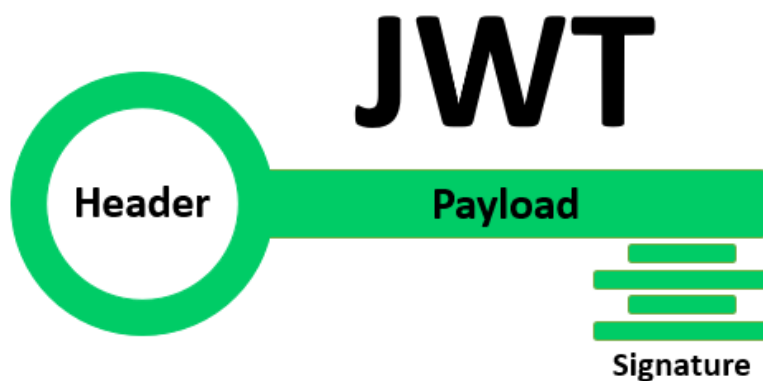


Рисунок 2.2 – Загальна структура JWT

Цей токен передається клієнтом разом із кожним запитом до сервера, що дає змогу серверу ідентифікувати користувача без необхідності постійного зберігання сесій чи запитів до бази даних. Крім того, токени можуть бути легко інтегровані з різними фронтенд- і бекенд-технологіями, забезпечуючи високу гнучкість у побудові масштабованої архітектури застосунку. Важливим аспектом JWT є те, що дані

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

всередині токена захищені цифровим підписом, тому будь-яка спроба змінити вміст токена буде одразу виявлена, що підвищує загальний рівень безпеки.

Для забезпечення інтуїтивно зрозумілої взаємодії з користувачем та покращення аналізу даних у системі також було впроваджено метод графічної візуалізації. Його суть полягає в тому, що дані, які зберігаються або генеруються під час роботи програми, представляються не лише у текстовому вигляді, а й за допомогою діаграм, графів, таблиць, візуальних компонентів інтерфейсу тощо. Зокрема, були реалізовані механізми візуального представлення категорій, станів завдань, зв'язків між об'єктами (наприклад, між категоріями й завданнями), а також графічне подання результатів авторизації й дій користувача в системі. Такий підхід значно покращує сприйняття інформації, спрощує навігацію, а також сприяє ефективному прийняттю рішень користувачами.

Для формування оптимального технологічного стеку використано метод порівняльного аналізу, що дозволив об'єктивно оцінити різні варіанти технологічних рішень за визначеними критеріями та зробити обґрунтований вибір інструментів для реалізації системи.

Основними критеріями оцінювання стали:

- продуктивність і масштабованість;
- швидкість розробки;
- безпека;
- сумісність;
- зручність підтримки та оновлення.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Таблиця 2.1 – Порівняльний аналіз технологій для реалізації вебзастосунку

Модуль	Варіанти технологій	Переваги	Недоліки	Обраний варіант
Frontend	React, Angular, Vue.js	React: висока продуктивність, компонентна архітектура, велика спільнота, активний розвиток, багатий набір бібліотек	Angular: складніший поріг входу Vue: менша кількість готових інтеграцій, слабша підтримка в корпоративному секторі	React
Backend	Node.js + Express, Django, Laravel	Node.js: асинхронна модель, швидке опрацювання запитів, підтримка REST API, зручна робота з JavaScript на клієнті й сервері	Django: обмежена гнучкість при розробці SPA Laravel: потребує більше ресурсів і часу на налаштування та розгортання	Node.js + Express
База даних	MongoDB, MySQL, PostgreSQL	MongoDB: гнучка структура, легка масштабованість, зручна інтеграція з Node.js, ідеально підходить для неструктурованих даних	MySQL/PostgreSQL: складніші при масштабуванні, жорстка схема	MongoDB
Аутентифікація	JWT, OAuth 2.0, Session-based auth	JWT: зручна передача в API, не потребує збереження стану	OAuth 2.0: складніша реалізація Session: потребує серверного зберігання	JWT

На основі аналізу було сформовано оптимальний технологічний стек, що дозволяє досягти поставленої мети. Таким чином, метод порівняльного аналізу дозволив обрати сучасні, сумісні між собою технології, які забезпечують ефективну реалізацію всіх функціональних модулів вебзастосунку.

2.2 Технології розробки системи

Результати порівняльного аналізу для реалізації вебзастосунку психологічної підтримки сформовано у сучасний технологічний стек, що охоплює всі необхідні компоненти системи.

ReactJS – це популярна бібліотека JavaScript, заснована на архітектурі компонентів, що дозволяє створювати динамічні, інтерактивні та масштабовані інтерфейси. Вона має велику спільноту та активно розвивається, забезпечуючи доступ до широкого набору готових бібліотек і рішень.

Node.js у поєднанні з фреймворком Express.js дозволяє створювати швидкі та масштабовані REST API. Node.js має асинхронну модель обробки запитів та забезпечує використання єдиної мови програмування JavaScript як на клієнтській, так і на серверній частинах.

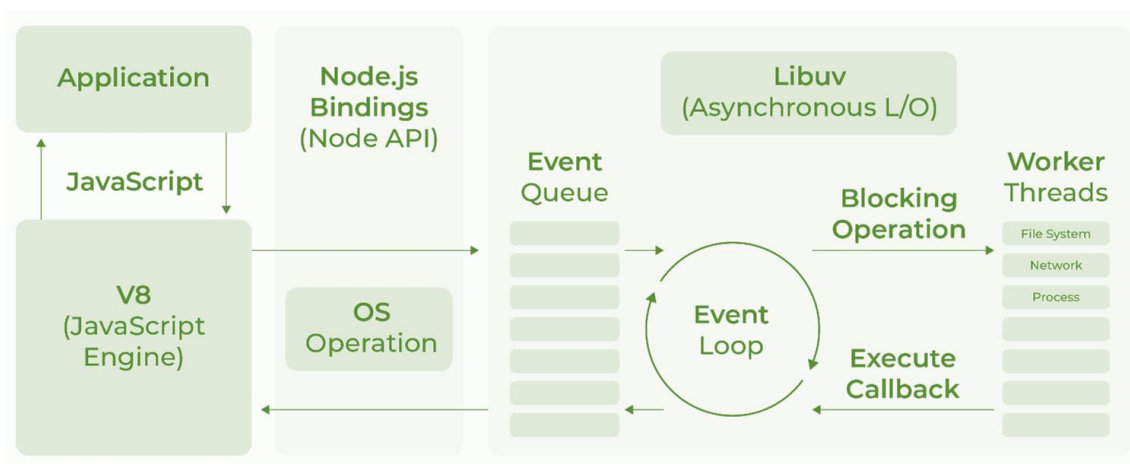


Рисунок 2.3 – Архітектура Node.js

MongoDB – це найвідоміша NoSQL-база даних, яка забезпечує ефективну інтеграцію з Node.js. MongoDB є оптимальним вибором для зберігання

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

неструктурованих та напівструктурованих даних, характерних для психологічних досліджень, повідомлень та записів користувачів.

Переваги використання MongoDB:

- підтримка спеціальних запитів;
- індексація;
- тиражування;
- агрегація;
- висока продуктивність;
- балансування навантаження;
- горизонтальна масштабованість;
- велике сховище медіа.

JWT – відкритий стандарт для створення токенів доступу, заснований на форматі JSON. Він дозволяє здійснювати безпечну аутентифікацію без збереження стану сесії на сервері, що є перевагою для розподілених систем.

Платформа Vercel забезпечує прості у використанні інструменти CI/CD, автоматичне масштабування, інтеграцію з GitHub та підтримку сучасних фреймворків. Vercel дозволяє швидко оновлювати систему та забезпечує високу доступність вебзастосунку.

Висновки до розділу 2

У другому розділі проведено детальний аналіз різних методів та технологій для розробки вебзастосунку. У результаті аналізу та порівняння можливих технологічних рішень для створення вебзастосунку психологічної підтримки було обрано оптимальний стек технологій, що відповідає вимогам продуктивності, безпеки, зручності розробки та масштабованості. Метод порівняльного аналізу дозволив

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

обґрунтовано оцінити переваги й недоліки сучасних інструментів, забезпечивши об'єктивний вибір.

Обрана комбінація технологій — React.js для frontend, Node.js + Express для backend, MongoDB для бази даних, JWT для аутентифікації та Vercel як платформи для хостингу забезпечує ефективну реалізацію функціональних модулів системи.

Розглянуто актуальні методи забезпечення безпеки даних, моделі автентифікації та авторизації користувачів, а також інструменти для графічної візуалізації, що відіграють важливу роль у побудові зручного інтерфейсу користувача.

Таким чином, теоретичні положення стали підґрунтям для формування загальної архітектури програмного продукту, визначення вимог до функціональності, безпеки та зручності інтерфейсу.

3 РОЗРОБКА ВЕБЗАСТОСУНКУ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

3.1 Аналіз вхідних даних та структура вебзастосунок

У цьому розділі здійснено детальний аналіз структури розробленої системи, типів вхідних даних, а також особливостей взаємодії між її компонентами. Система реалізована за допомогою сучасного веб-стека технологій: React.js використовується для створення клієнтської частини, Node.js разом з Express – для побудови серверної логіки, MongoDB – як гнучке NoSQL-сховище даних, а JWT – для реалізації механізму безпечної аутентифікації та авторизації користувачів.

Система обробляє різноманітні вхідні дані, що надходять переважно від користувача через клієнтський інтерфейс, а також отримуються із зовнішніх запитів до серверної частини. Основні типи вхідних даних:

- облікові дані користувача (логін та пароль). Паролі передаються у зашифрованому вигляді через захищене з'єднання та обробляються на сервері з використанням алгоритму хешування bcrypt;
- користувацькі записи. Дані включають текстові поля, дати, статуси, категорії тощо;
- HTTP-запити (GET, POST, PUT, DELETE). Використовуються для взаємодії REST API;
- дані сесії та автентифікації.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Таблиця 3.1 – Приклади вхідних даних

Поле	Тип	Опис
email	рядок	Електронна адреса користувача
password	рядок (хеш)	Пароль користувача у вигляді хешу
token	JWT-токен	Авторизаційний токен
title	рядок	Назва нотатки
description	рядок	Текстовий опис
categoryId	ObjectId	Посилання на категорію
createdAt	дата	Дата створення

Архітектура розробленої системи побудована відповідно до принципів клієнт-серверної моделі з розділенням відповідальності.

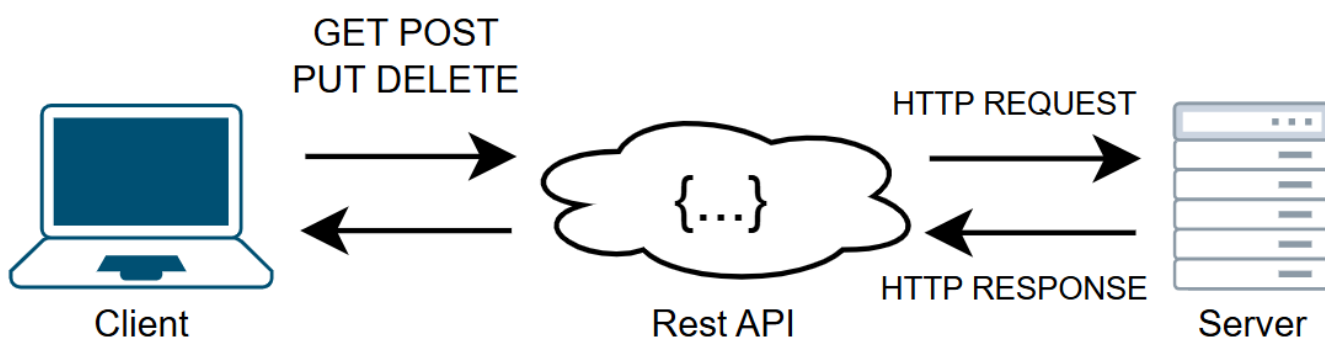


Рисунок 3.1 – Принцип роботи REST API

Принцип роботи REST API збігається із загальним принципом роботи в Інтернеті з будь-яким сайтом. Користувач (клієнт) зв'язується із сервером за

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

допомогою API, коли йому потрібний будь-який веб-ресурс. Даний запит обробляється і клієнт отримує запитуваний ресурс.

Поетапно роботу REST API можна описати наступним чином.

1. Клієнт надсилає запит на сервер. Згідно з документацією API, клієнт форматує запит таким чином, щоб сервер отримав відомості, необхідні для обробки.
2. Сервер автентифікує клієнта та підтверджує, що клієнт має право зробити цей запит. Якщо аутентифікація пройшла неуспішно, помилка 403 повертається.
3. Сервер отримує запит у потрібному вигляді, обробляє його.
4. Потім сервер повертає відповідь на запит клієнта. У відповіді міститься інформація про успішність виконання команди та запитуваний ресурс.

Для роботи з ресурсами через REST API використовуються команди HTTP.

1. GET для отримання інформації про дані: на запит GET /users ми отримаємо список усіх користувачів.
2. DELETE для видалення даних: на запит DELETE /iemail буде видалений користувач із заданою поштою.
3. POST для додавання нових або заміни старих даних: запит POST /user додасть нового користувача.
4. PUT для редагування даних: за запитом PUT /users можна внести редагування до списку користувачів, представлених у додатку.

3.2 Проєктування та моделювання вебзастосунку служби психологічної підтримки

Для ознайомлення зі сценаріями використання вебзастосунку створимо діаграми варіантів використання та опишемо їх за допомогою сценарної техніки опису взаємодії Use Case.

Для створення діаграм варіантів використання обрано вебзастосунок для

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

розробки блок-схем, діаграм, UML та інших під назвою draw.io [10].

UML-діаграми варіантів використання (Use Case diagrams) — це один із типів діаграм, що входить до мови моделювання UML. Вони дозволяють візуально зобразити взаємодію користувачів (акторів) із системою, відображаючи функціональні можливості, які надає система. Такі діаграми допомагають зрозуміти вимоги до системи, визначити ролі користувачів та їх цілі.

На діаграмах представлені три основні актори.

1. Клієнт – особа, яка потребує психологічної підтримки.
2. Психолог – фахівець, що надає консультації.
3. Адміністратор – особа, що здійснює управління системою.

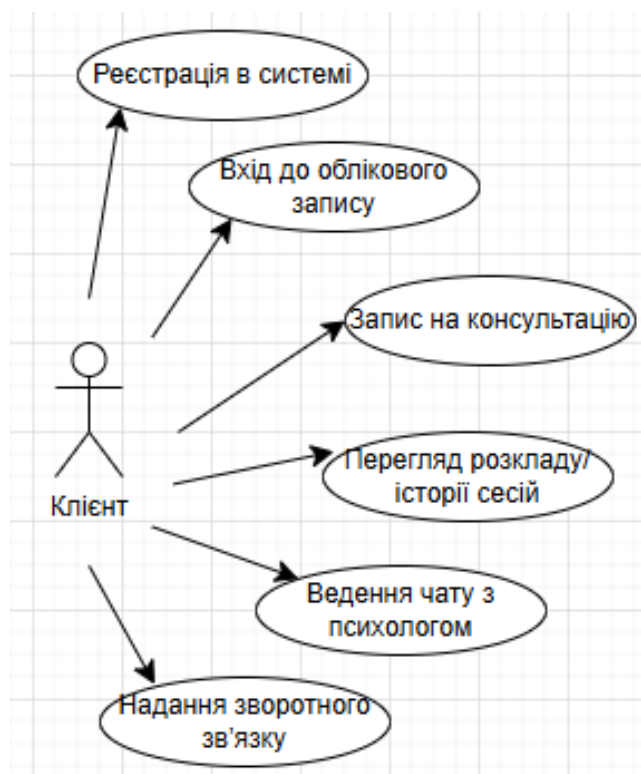


Рисунок 3.2 – Діаграма варіантів використання для Клієнта

За допомогою діаграми напишемо сценарій до кожного варіанту використання

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

для Клієнта.

1. Реєстрація в системі.

Передумови: користувач не має облікового запису.

Основний сценарій:

- клієнт відкриває головну сторінку вебсайту;
- обирає опцію "Зареєструватися";
- вводить необхідну інформацію (ім'я, електронну пошту, пароль);
- підтверджує реєстрацію;
- система надсилає листа для підтвердження на email;
- клієнт переходить за посиланням у листі – реєстрація завершується.

2. Вхід до облікового запису.

Передумови: користувач вже зареєстрований.

Основний сценарій:

- клієнт відкриває головну сторінку вебсайту;
- обирає опцію «Увійти»;
- вводить необхідну інформацію (електронну пошту, пароль);
- система перевіряє дані;
- у разі успішної перевірки відкривається особистий кабінет.

Альтернативний варіант:

- якщо дані введено некоректно, то система виводить повідомлення про помилку.

3. Запис на консультацію.

Передумови: клієнт авторизований у системі.

Основний сценарій:

- клієнт переходить до розділу «Запис на консультацію»;
- обирає психолога зі списку;

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

- переглядає доступні дати й години;
- обирає зручний час;
- підтверджує запис;
- система створює подію в календарі психолога.

4. Перегляд розкладу/історії сесії.

Передумови:

- клієнт авторизований у системі;
- у клієнта є заплановані або вже проведені консультації.

Основний сценарій:

- клієнт входить у свій особистий кабінет;
- обирає розділ «Мої сесії»;
- система відображає список майбутніх та минулих сесій із зазначенням

необхідної інформації (дата та час, ім'я психолога, статус).

Альтернативний варіант:

- якщо клієнт не має жодної сесії, то система показує повідомлення: «У вас поки немає запланованих/завершених сесій».

5. Ведення чату з психологом.

Передумови:

- клієнт авторизований у системі;
- існує підтверджена консультація або психолог дозволяє відкрити чат.

Основний сценарій:

- клієнт переходить у розділ «Чати»;
- обирає психолога, з яким хоче поспілкуватися;
- система відкриває чат-інтерфейс;
- клієнт вводить повідомлення та натискає «Надіслати»;

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

- повідомлення зберігається в системі й миттєво надсилається психологу;
- психолог відповідає у тому ж чаті (в реальному часі).

Альтернативний варіант:

- якщо чат недоступний, то система виводить повідомлення: «Чат тимчасово недоступний».

6. Надання зворотного зв'язку.

Передумови:

- клієнт авторизований у системі;
- консультація з психологом відбулася (статус сесії – «завершено»);
- час, відведений на залишення відгуку, ще не сплив.

Основний сценарій:

- клієнт входить у розділ «Мої сесії»;
- серед завершених сесій клієнт обирає ту, щодо якої хоче залишити відгук;
- система відкриває форму зворотного зв'язку;
- клієнт заповнює форму та натискає кнопку "Надіслати";
- система зберігає зворотний зв'язок, повідомляє клієнта про успішну відправку та додає відгук до публічного профілю психолога.

Альтернативний варіант:

- якщо клієнт уже залишав відгук по цій сесії, то система блокує повторну відправку і виводить повідомлення: «Ви вже залишили відгук про цю сесію».

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

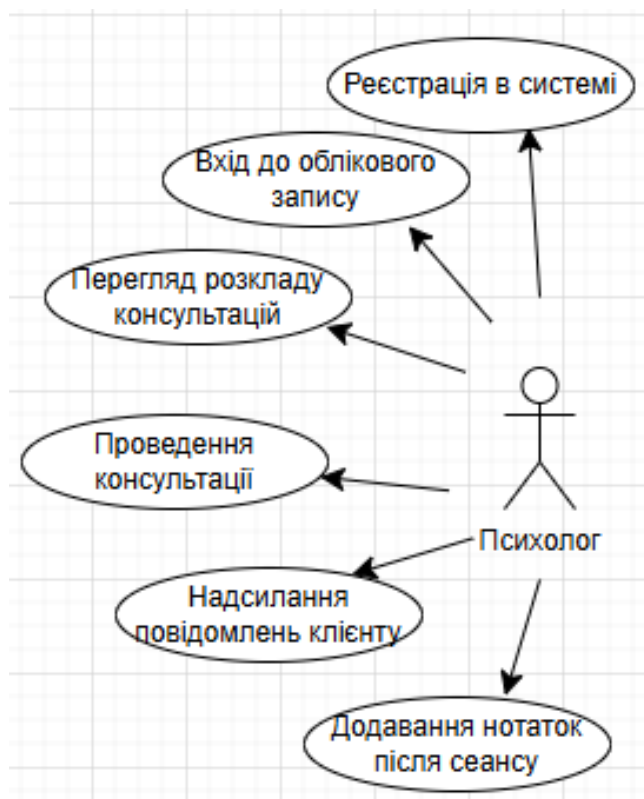


Рисунок 3.3 – Діаграма варіантів використання для Психолога

За допомогою діаграми напишемо сценарій до кожного варіанту використання для Психолога.

1. Реєстрація в системі.

Передумови: психолог не має облікового запису.

Основний сценарій:

- психолог відкриває головну сторінку вебсайту;
- обирає опцію "Зареєструватися";
- вводить необхідну інформацію;
- підтверджує реєстрацію;
- система надсилає листа для підтвердження на email;
- психолог переходить за посиланням у листі – реєстрація завершується.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

2. Вхід до облікового запису.

Передумови: психолог вже зареєстрований.

Основний сценарій:

- психолог відкриває головну сторінку вебсайту;
- обирає опцію «Увійти»;
- вводить необхідну інформацію (електронну пошту, пароль);
- система перевіряє дані;
- у разі успішної перевірки надається доступ до персонального кабінету.

Альтернативний варіант:

- якщо дані введено некоректно, то система виводить повідомлення про помилку.

3. Перегляд розкладу консультацій.

Передумови:

- психолог авторизований у системі;
- є заплановані консультації.

Основний сценарій:

- психолог переходить до розділу «Мій розклад»;
- система відображає календар зі списком майбутніх консультацій;
- психолог може змінити або скасувати сесію.

Альтернативний варіант:

- якщо у психолога немає жодної сесії, то система показує повідомлення «Розклад порожній».

4. Проведення консультацій.

Передумови:

- психолог авторизований у системі;
- запланована сесія та обидва учасники онлайн.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Основний сценарій:

- психолог натискає «Почати консультацію» у розкладі;
- система відкриває відеозв'язок з клієнтом;
- відбувається онлайн-консультація;
- після завершення сесії сеанс закривається, зберігається в історії.

Альтернативний варіант:

- якщо клієнт не з'явився, то психолог може запропонувати інший час;
- якщо клієнт перервав сеанс, то система зберігає неповну консультацію в історії.

5. Надсилення повідомлень клієнту.

Передумови:

- психолог авторизований у системі;
- запланована сесія або вже є активний чат.

Основний сценарій:

- психолог заходить до чату з клієнтом;
- система відкриває чат-інтерфейс;
- вводить повідомлення та натискає «Надіслати»;
- повідомлення зберігається в системі й миттєво надсилається клієнту;
- клієнт бачить повідомлення та відповідає у тому ж чаті (в реальному часі).

Альтернативний варіант:

- якщо чат недоступний, то система виводить повідомлення: «Чат тимчасово недоступний».

6. Додавання нотаток після сеансу.

Передумови:

- психолог авторизований у системі;

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

- сеанс завершено.

Основний сценарій:

- психолог відкриває завершену сесію з клієнтом;
- натискає «Додати нотатку»;
- вводить текстовий опис;
- система зберігає нотатку, доступну лише психологу.

Альтернативний варіант:

- якщо психолог випадково закриває вкладку, то дані не зберігаються;
- обмеження за обсягом тексту.

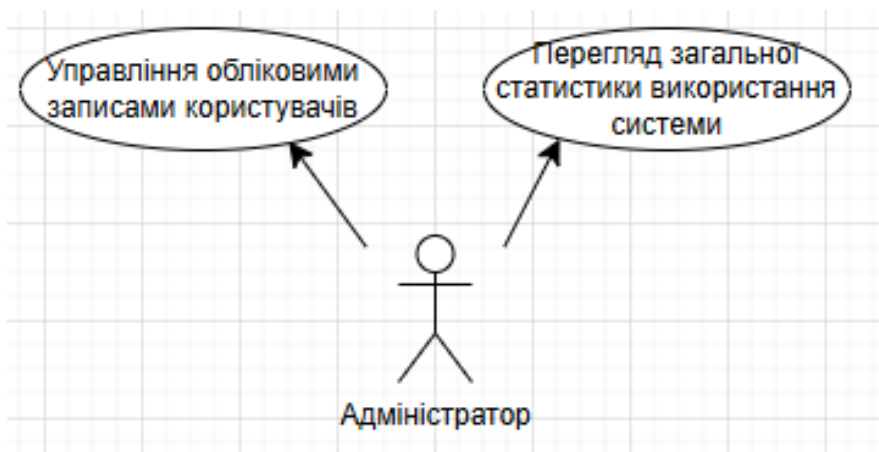


Рисунок 3.4 – Діаграма варіантів використання для Адміністратора

За допомогою діаграми напишемо сценарій до кожного варіанту використання для Адміністратора.

1. Управління обліковими записами користувачів.

Основний сценарій:

- адміністратор входить у систему;
- обирає розділ «Користувачі»;

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

- переглядає список користувачів;
- обирає конкретного користувача;
- виконує одну з дій: активувати/деактивувати обліковий запис, видалити користувача, переглянути історію активності.

Альтернативний варіант:

- якщо виникає помилка при збереженні, то дія не виконується, адміністратор бачить помилку.

2. Перегляд загальної статистики використання системи.

Основний сценарій:

- адміністратор входить у систему;
- обирає розділ «Статистика»;
- переглядає загальну аналітику: кількість зареєстрованих клієнтів/психологів, кількість проведених консультацій, активність користувачів.

Альтернативний варіант:

- якщо виникає помилка при завантаженні даних, то система пропонує спробувати пізніше.

Загальну діаграму варіантів використання для акторів можна переглянути в Додатку А.

Діаграма колекцій відіграє ключову роль у процесі проєктування об'єктно-орієнтованої системи. Вона застосовується на різних стадіях розробки — від початкового планування до реалізації — з різним рівнем деталізації.

В конкретному випадку, вебзастосунок складається з 7 колекцій. Діаграма колекцій для цього вебзастосунку представлена на рис. 3.4.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

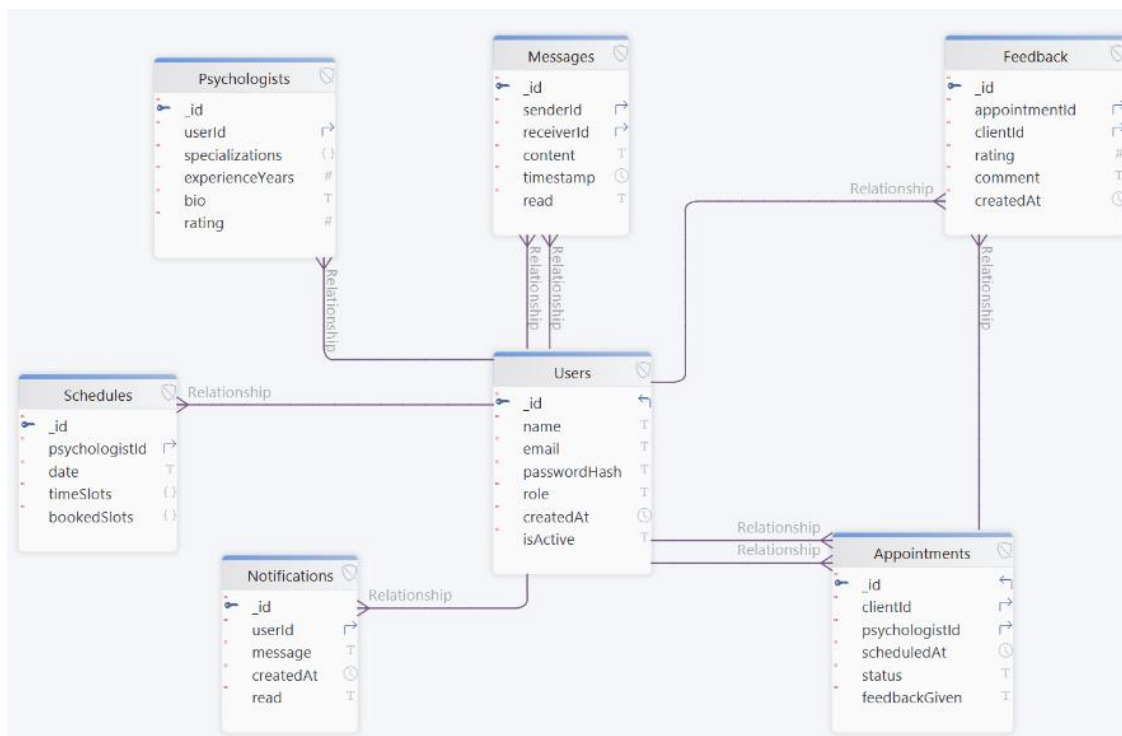


Рисунок 3.5 – Діаграма колекцій вебзастосунку

Колекції створенні за допомогою бази даних типу NoSQL, що відрізняються від традиційних реляційних баз, оскільки вони ґрунтуються на інших моделях зберігання й обробки даних.

Для проєктування інформаційної системи використовується редактор коду Visual Studio Code [14]. Цей інструмент є одним із найпопулярніших серед розробників завдяки своїй гнучкості, широким можливостям розширення та підтримці багатьох мов програмування. Visual Studio Code забезпечує інтеграцію з системами контролю версій, такими як Git, що дозволяє ефективно керувати змінами в проєкті та організовувати командну роботу.

Крім того, VS Code підтримує велику кількість розширень, які полегшують роботу з базами даних (наприклад, MongoDB), а також інструменти для візуалізації структури даних, автодоповнення, підсвічування синтаксису, дебагінг і тестування.

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

Завдяки цьому розробники можуть створювати, редагувати та відлагоджувати код без необхідності перемикатися між різними середовищами.

У процесі проектування системи також можна використовуватись додаткові плагіни для перевірки якості коду та забезпечення його відповідності сучасним стандартам. Таким чином, Visual Studio Code виконує роль централізованого середовища розробки, яке поєднує всі необхідні етапи створення інформаційної системи — від проектування до реалізації та тестування.

Для реалізації серверної частини системи обрано платформу Node.js у поєднанні з фреймворком Express.js, що забезпечує підвищену продуктивність і спрощує розробку RESTful API. Такий технологічний стек дозволяє створювати масштабовані, неблокуючі веб-сервіси з високою швидкістю обробки запитів.

Express.js — це мінімалістичний та водночас потужний фреймворк для Node.js, який дозволяє швидко створювати серверні веб-додатки та API. Він забезпечує зручні інструменти для маршрутизації HTTP-запитів, використання middleware-функцій, обробки помилок, а також роботи з шаблонами сторінок, якщо це потрібно.

Express має модульну архітектуру, яка дозволяє легко підключати сторонні пакети — наприклад, для реалізації авторизації (через бібліотеки на кшталт Passport.js або jsonwebtoken), валідації даних (наприклад, express-validator), логування (morgan), обробки запитів (body-parser або multer для файлів), а також взаємодії з базами даних (наприклад, Mongoose для MongoDB або Sequelize для SQL-орієнтованих БД).

Для документування API використовується інструмент Postman, що дозволяє іншим розробникам легко ознайомитися з доступними маршрутами, параметрами запитів та відповідями.

Висновки до розділу 3

У третьому розділі було здійснено комплексну розробку інформаційної системи, що охоплює як технічну реалізацію, так і аналіз її структурних компонентів. На основі використання сучасного веб-стека технологій – React.js, Node.js, Express.js, MongoDB та JWT – створено повноцінну клієнт-серверну архітектуру, яка відповідає актуальним вимогам до безпеки, масштабованості та гнучкості програмних рішень.

Проаналізовано типи вхідних даних, які надходять як від користувачів, так і із зовнішніх джерел. Наведено приклади використання HTTP-запитів, які реалізують CRUD-функціональність у межах системи. Завдяки застосуванню токенів доступу (JWT) забезпечено надійний рівень автентифікації та авторизації користувачів.

Особливу увагу приділено етапу моделювання системи. За допомогою діаграм варіантів використання (Use Case) сформульовано основні сценарії взаємодії між користувачами та системою: клієнтом, психологом і адміністратором. Діаграми колекцій відобразили внутрішню структуру зберігання даних у базі MongoDB, що дозволило оптимізувати запити та уніфікувати роботу з колекціями.

Проектування проводиться з використанням Visual Studio Code як основного середовища розробки, що забезпечило гнучке керування кодом, зручну інтеграцію з Git та підтримку численних розширень.

Для реалізації серверної частини обрано Node.js з фреймворком Express.js, що дозволило реалізувати високопродуктивний REST API. Розроблена система є надійною основою для впровадження у сфері психологічних онлайн-консультацій.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА ДОКУМЕНТАЦІЇ

4.1 Опис програмної реалізації

Для розробки застосунку дуже важливо правильно налаштувати середовище для програмування, в нашому випадку Visual Studio Code. Для початку запускається саме середовище, після чого відкривається головне вікно редактору з фрагментом гарячих клавіш по роботі з кодом (рис. 4.1).

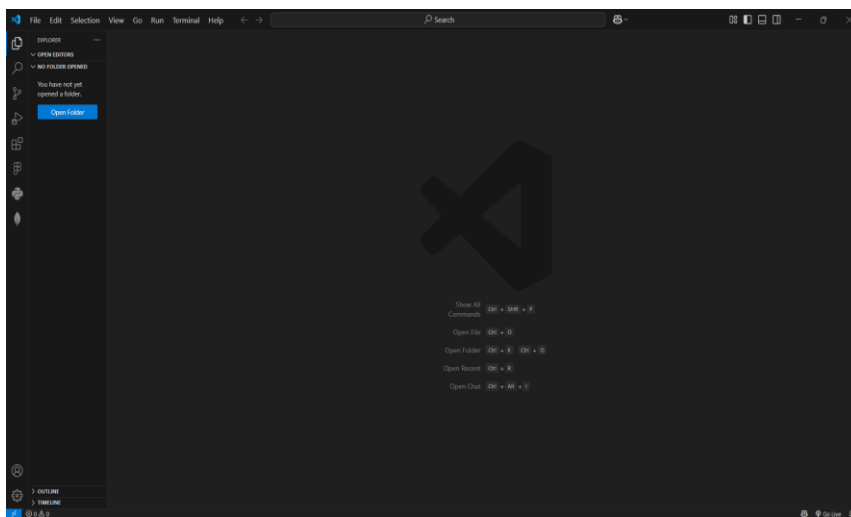


Рисунок 4.1 – Головне вікно редактору

Для зручності та практичності в програмуванні середовище також пропонує завантажити додаткові розширення, в нашому випадку будуть доречні наступні плагіни.

1. HTML/CSS/JavaScript Snippets – для швидкого вставлення та повторного використання готових фрагментів коду.
2. MongoDB for VS Code – для роботи з базою даних MongoDB прямо з редактора Visual Studio Code, без необхідності відкривати сторонні інтерфейси чи термінали.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

3. Prettier - Code formatter – для автоматичного форматування коду.
4. Reactjs code snippets – для швидкої розробки, додаючи готові фрагменти коду (сніпети) з використанням React.js.

Після запуску та перевірки роботи стартової сторінки відкриваємо проєкт у Visual Studio Code. Для цього достатньо обрати каталог з проєктом у файловому менеджері. Після відкриття у VSCode у лівій панелі буде відображено повну структуру проєкту (рис. 4.2).

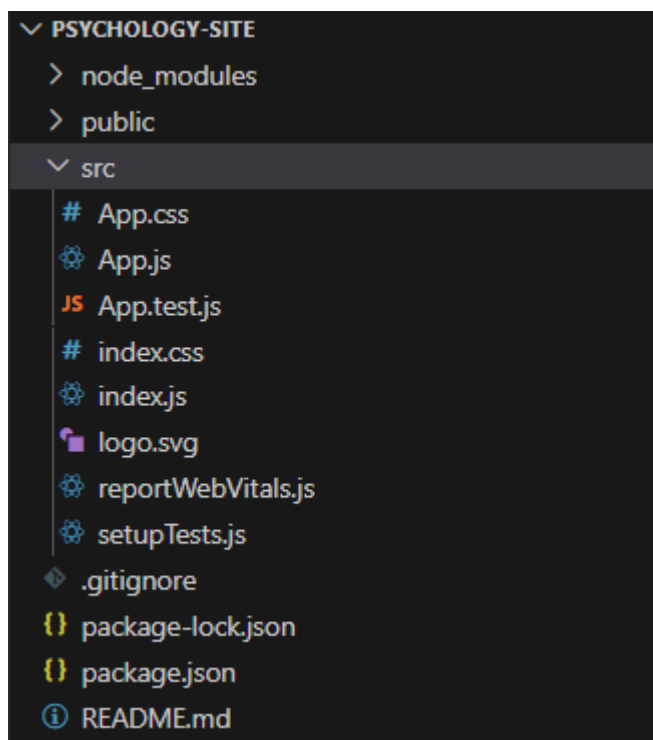


Рисунок 4.2 – Повна структура проєкту

Далі необхідно створити нову директорію для файлів backend та в ній підключити базу даних MongoDB. Потрібно додатково встановити бібліотеку mongoose командою `npm install mongoose`, а у файлі серверної частини додати підключення:

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

```
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
const app = express();
app.use(cors({
  origin: 'http://localhost:3000',
  credentials: true
}));
app.use(express.json());
mongoose.connect('mongodb+srv://nkudrya2004:VdZCvxBBNywppNrY@cluster0.81czigz.mongoddb.net/Breathe_freely?retryWrites=true&w=majority', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})
  .then(() => console.log('Connected to MongoDB'))
  .catch((err) => console.error('MongoDB error:', err));
app.listen(5000, () => {
  console.log('Сервер запущено на http://localhost:5000');
});
```

Повний код реалізації представлено в Додатку В.

Щоб запустити сервер, потрібно в командному рядку перейти до директорії, в нашому випадку «server», в якій розташований файл серверу, в нашому випадку «server.js», та за допомогою команди `node server.js` запустити сервер (рис. 4.3).

```
PS C:\Users\ndzdk\OneDrive\Рабочий стол\Диплом\psychology-site> cd server
PS C:\Users\ndzdk\OneDrive\Рабочий стол\Диплом\psychology-site\server> node server.js
Сервер запущено на http://localhost:5000
Connected to MongoDB
█
```

Рисунок 4.3 – Запуск серверної частини

З рисунку 4.3 бачимо, що консоль вивела два рядки: перший рядок «Сервер запущено на `http://localhost:5000`» свідчить про те, що вдалося запустити сервер на порту: 5000; другий рядок «Connected to MongoDB» свідчить про те, що вдалося серверу підключитися до MongoDB.

На початку розробки розглянемо у нашому проєкті файл `package.json`. Це головний конфігураційний файл проєкту. Він містить інформацію про сам проєкт,

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

його залежності, скрипти для запуску, авторів, ліцензію тощо.

Приклад package.json:

```
{
  "name": "psychology-site",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/dom": "^10.4.0",
    "@testing-library/jest-dom": "^6.6.3",
    "@testing-library/react": "^16.3.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^19.1.0",
    "react-dom": "^19.1.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

Клієнтська частина, як було обговорено раніше, базується на фреймворку React та використовує мову програмування JavaScript.

При потраплянні на головну сторінку сайту, користувач зустрічає варіант увійти до системи.

Увійшовши до системи, користувач може отримати доступ до особистого

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

профілю. Щоб авторизуватися користувачу необхідно мати вже створений акаунт та ввести облікові дані (пошту та пароль) (рис. 4.4).

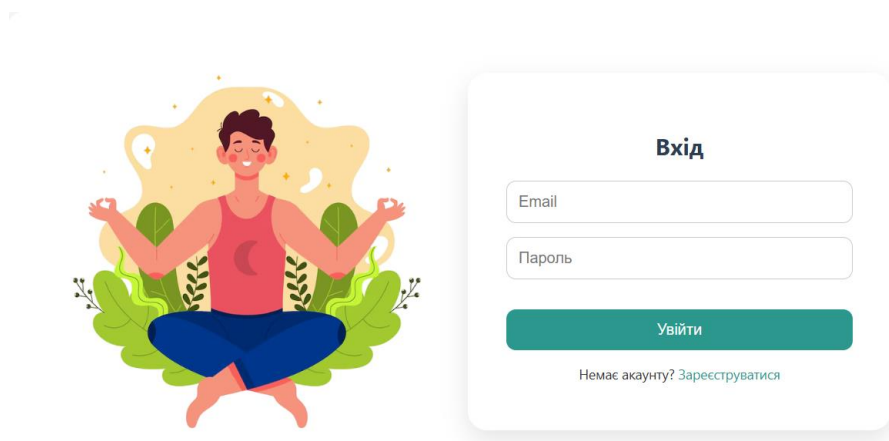


Рисунок 4.4 – Вікно авторизації користувача

Якщо користувач не має створеного акаунту, то натиснувши на «Зареєструватися», користувач переходить на сторінку реєстрації (рис. 4.5), де вводить свої дані, такі як ім'я, прізвище, дата народження, електронна пошта, пароль та підтвердження пароля.

Після заповнення даних та натискання на кнопку «Зареєструватися», система виконує перевірку введених даних, а саме перевірка унікальності електронної пошти або надійність пароля. У разі успішної перевірки, система реєструє користувача та перенаправляє на сторінку авторизації.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб



Реєстрація

Ім'я

Прізвище

☐ Жінка ☐ Чоловік

ДД.ММ.ГГГГ

Email

Пароль

☐ Я клієнт ☐ Я психолог

Зареєструватися

Вже маєте акаунт? Увійти

Рисунок 4.5 – Вікно реєстрації користувача

У програмному коді додатково реалізовано механізми перевірки введених користувачем даних та авторизації. Валідація забезпечує перевірку коректності полів згідно з визначеними правилами: перевіряється наявність значення, відповідність формату, мінімальна та максимальна довжина тощо. Це дозволяє запобігти некоректному введенню даних.

Окрім цього, код містить логіку авторизації користувачів. Вона включає перевірку логіна та пароля, а також автентифікацію через ролі, сесії або токени. Авторизованим користувачам надається доступ до захищених функцій або ресурсів системи, тоді як неавторизовані запити блокуються або обмежуються.

Фрагмент коду для валідації даних:

```
function isValidEmail(email) {
    const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return re.test(email);
}

function isValidPassword(password) {
    return password.length >= 6;
}

if (!isValidEmail(formData.email)) {
    setErrorMessage("Невірний формат email");
    return;
}
```

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

```
if (!isValidPassword(formData.password)) {
    setErrorMessage("Пароль повинен містити щонайменше 6 символів");
    return;
}
```

Фрагмент коду для авторизації користувача:

```
const { email, password } = req.body;

try {
    const user = await User.findOne({ email });
    if (!user) {
        return res.status(404).json({ message: 'Користувача не знайдено' });
    }
    const isMatch = await bcrypt.compare(password, user.password);
    if (!isMatch) {
        return res.status(401).json({ message: 'Невірний пароль' });
    }
    const token = jwt.sign(
        { id: user._id, email: user.email, role: user.role },
        JWT_SECRET,
        { expiresIn: '7d' }
    );
    res.json({ token });
} catch (error) {
    console.error('Login error:', error);
    res.status(500).json({ message: 'Помилка сервера при вході' });
}
```

4.2 Структура проєкту

При завантаженні сайту для всіх користувачів доступна головна сторінка застосунку (рис. 4.6), що є вітальним екраном, яка знайомить користувача з місією сервісу та пропонує основні функції платформи психологічної підтримки. Вона має кілька змістових секцій: вступний блок із заголовком, описом і кнопкою початку консультації; візуальний блок із зображенням для емоційного залучення; секцію з переліком напрямків консультування (рис. 4.7), де представлені ключові теми, з якими працюють психологи; блок із презентацією команди спеціалістів (рис. 4.8), що динамічно підтягується з сервера; інтерактивний калькулятор емоційного стану (рис. 4.9), який дозволяє оцінити свій психоемоційний стан за шкалою; а також розділ з відповідями на найчастіші запитання (рис. 4.10).

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

Після прийняття рішення про консультацію, користувач переходить до процесу запису до психолога. На цьому етапі спочатку обирається зручна дата та час сеансу серед доступних варіантів. Після вибору часу користувач перенаправляється до платіжної системи LiqPay для здійснення оплати. Після успішної транзакції відповідна сесія автоматично додається до особистого кабінету користувача, звідки можна переглянути деталі зустрічі та, за потреби, внести зміни або скасувати запис згідно з умовами сервісу. Реалізацію інтеграції платіжної системи LiqPay для онлайн-оплат консультацій наведено у Додатку Д.

Дизайн орієнтований на доброзичливість, доступність і довіру, а структура сторінки допомагає користувачу швидко зорієнтуватися у можливостях сервісу.

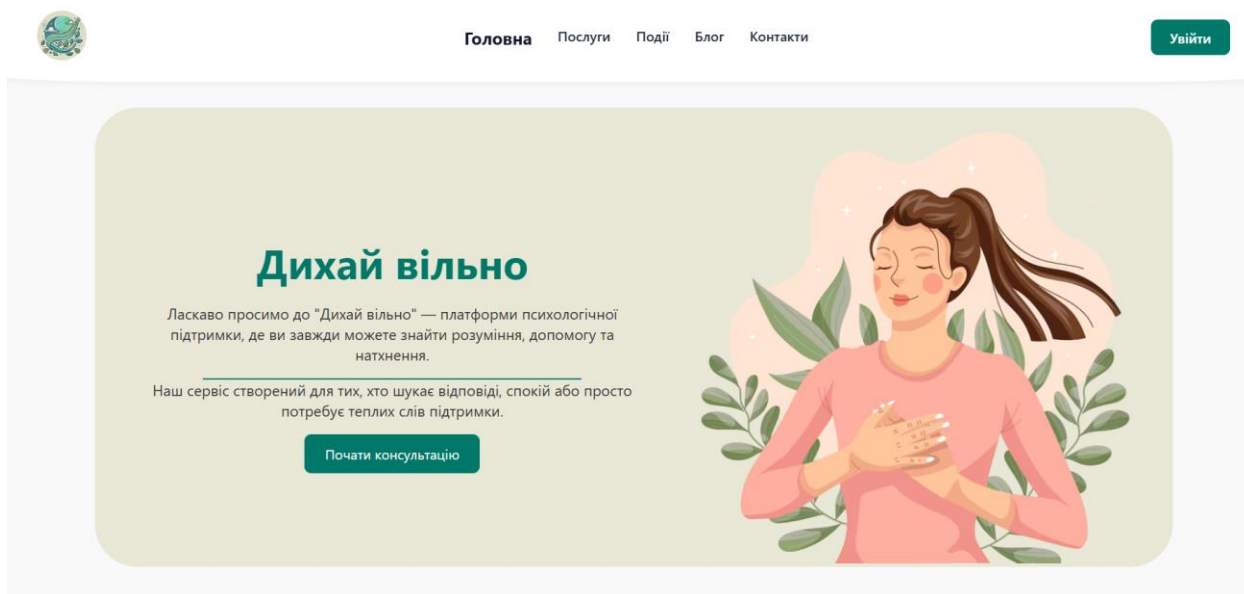


Рисунок 4.6 – Головна сторінка вебзастосунку

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

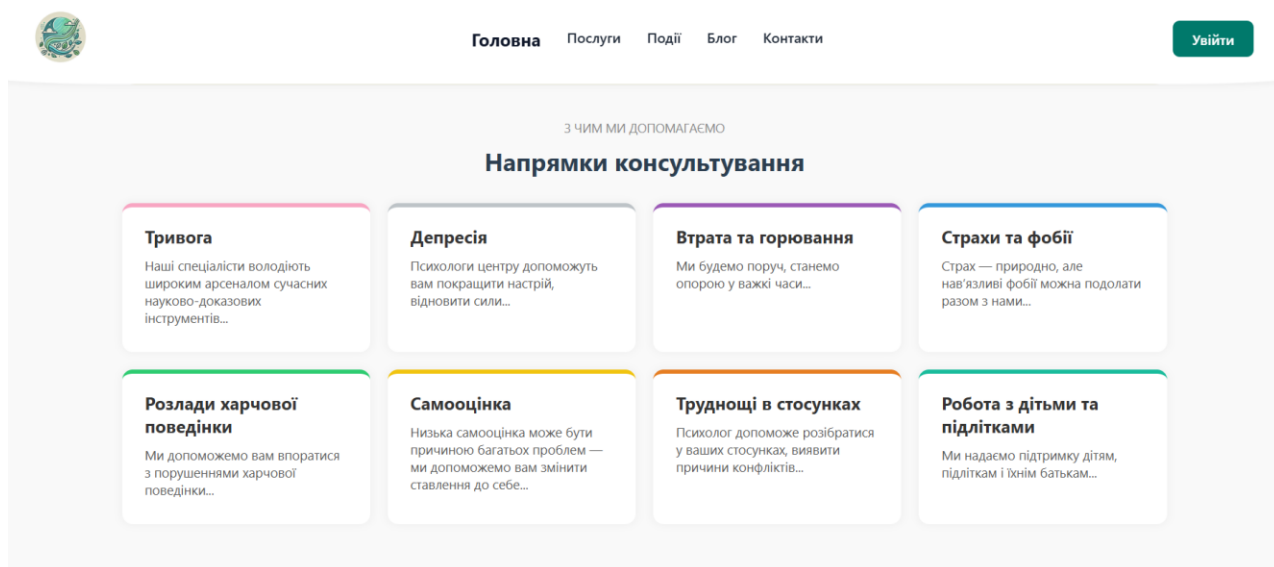


Рисунок 4.7 – Секція з переліком напрямків консультування

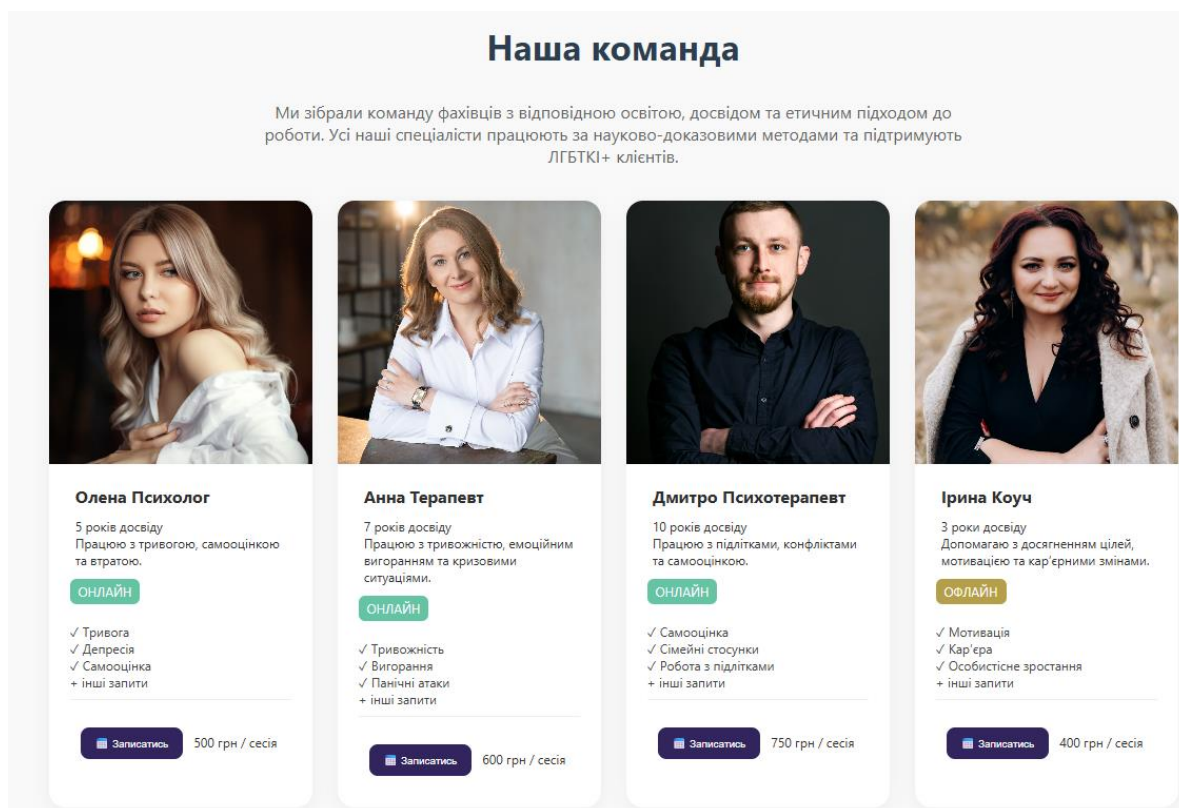


Рисунок 4.8 – Секція з переліком команди спеціалістів

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

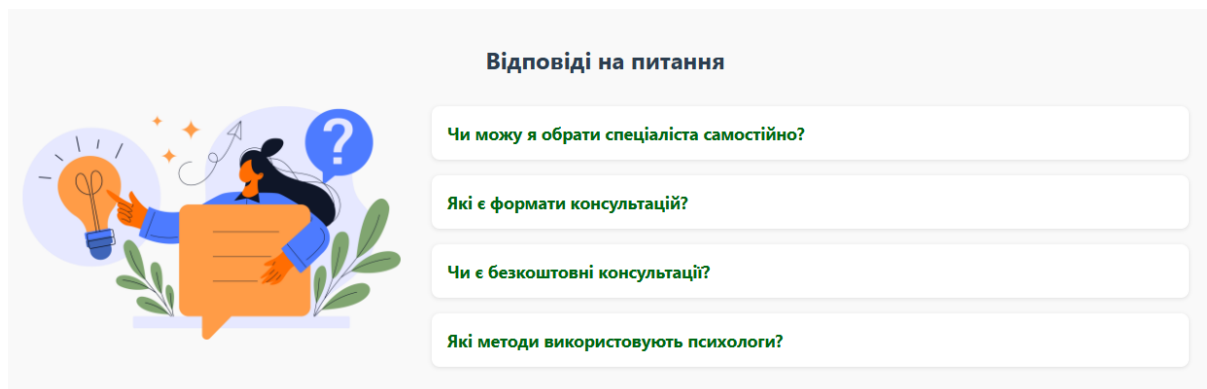


Рисунок 4.9 – Розділ з відповідями на найчастіші запитання

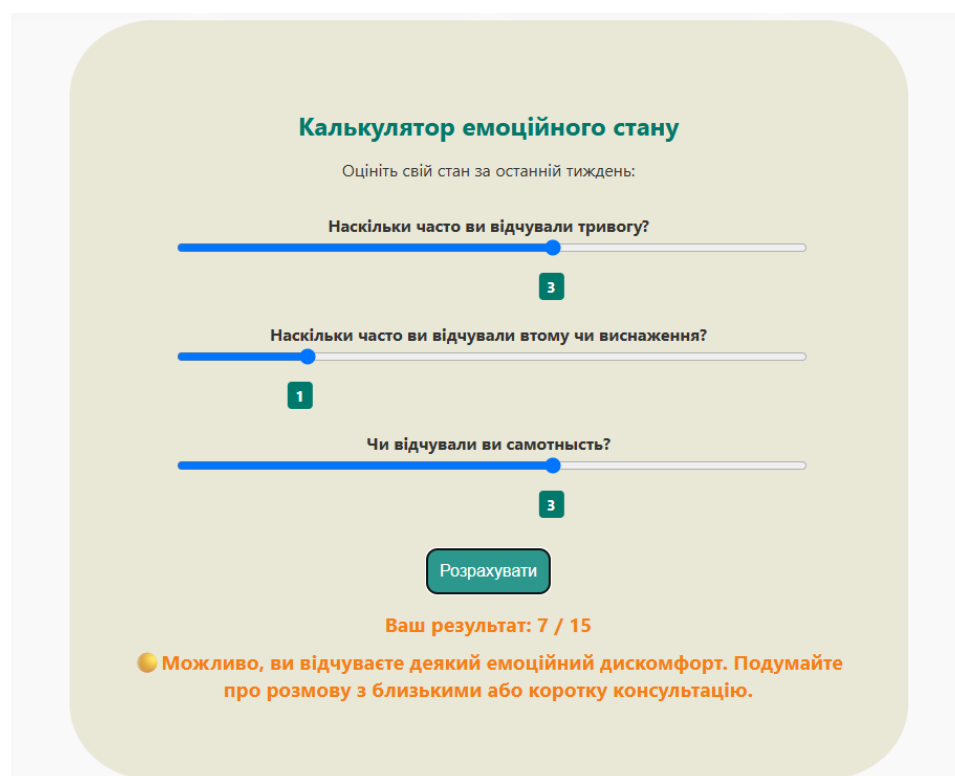


Рисунок 4.10 – Інтерактивний калькулятор емоційного стану

Після знайомства з головною сторінкою користувач може детальніше ознайомитись з доступними напрямками допомоги на сторінці «Послуги». Вона представляє структурований перелік форматів психологічної підтримки (рис. 4.11),

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

які надає платформа, у вигляді сітки карток. Кожна картка містить іконку, назву послуги та короткий опис, що допомагає швидко зорієнтуватися у різноманітті доступних варіантів.

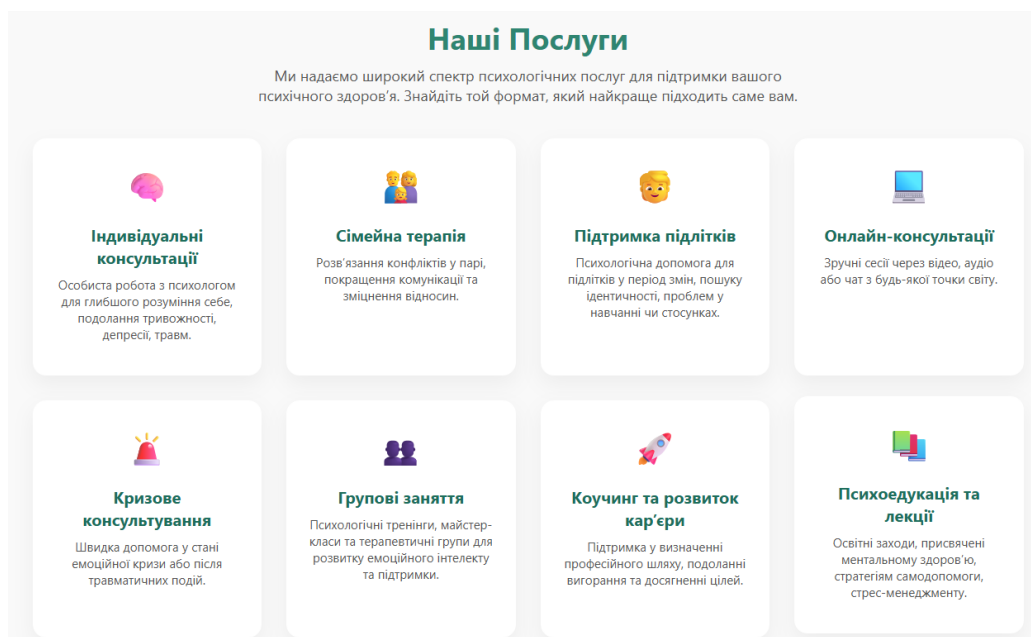


Рисунок 4.11 – Перелік форматів психологічної підтримки

Серед запропонованих напрямків – індивідуальні консультації, сімейна терапія, підтримка підлітків, кризове консультування, коучинг, психоедукація та інші. Такий підхід дозволяє охопити різні потреби користувачів: від вирішення особистих проблем до розвитку кар'єри чи проходження через складні життєві ситуації.

Завдяки чіткій візуальній ієрархії, дружньому дизайну й іконкам, що асоціюються з тематикою кожної послуги, сторінка є зручною для сприйняття і підкреслює відкритість сервісу до різних аудиторій.

У нижній частині сторінки інтегровано компонент Record, який пропонує пройти коротке опитування або обрати спеціаліста. Це сприяє швидкому переходу від ознайомлення до дії, що важливо в умовах, коли людина шукає допомогу.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Наступною сторінкою, з якою користувач може ознайомитися, є «Події». Вона розкриває соціально-активну складову платформи психологічної підтримки та слугує простором для інформування користувачів про заплановані заходи, минулі ініціативи та відгуки учасників.

Актуальні події (рис. 4.12) відображаються у вигляді карток із назвою, датою, банером, коротким описом та кнопкою для реєстрації. Такий підхід дозволяє легко зорієнтуватися в тому, що відбуватиметься найближчим часом.

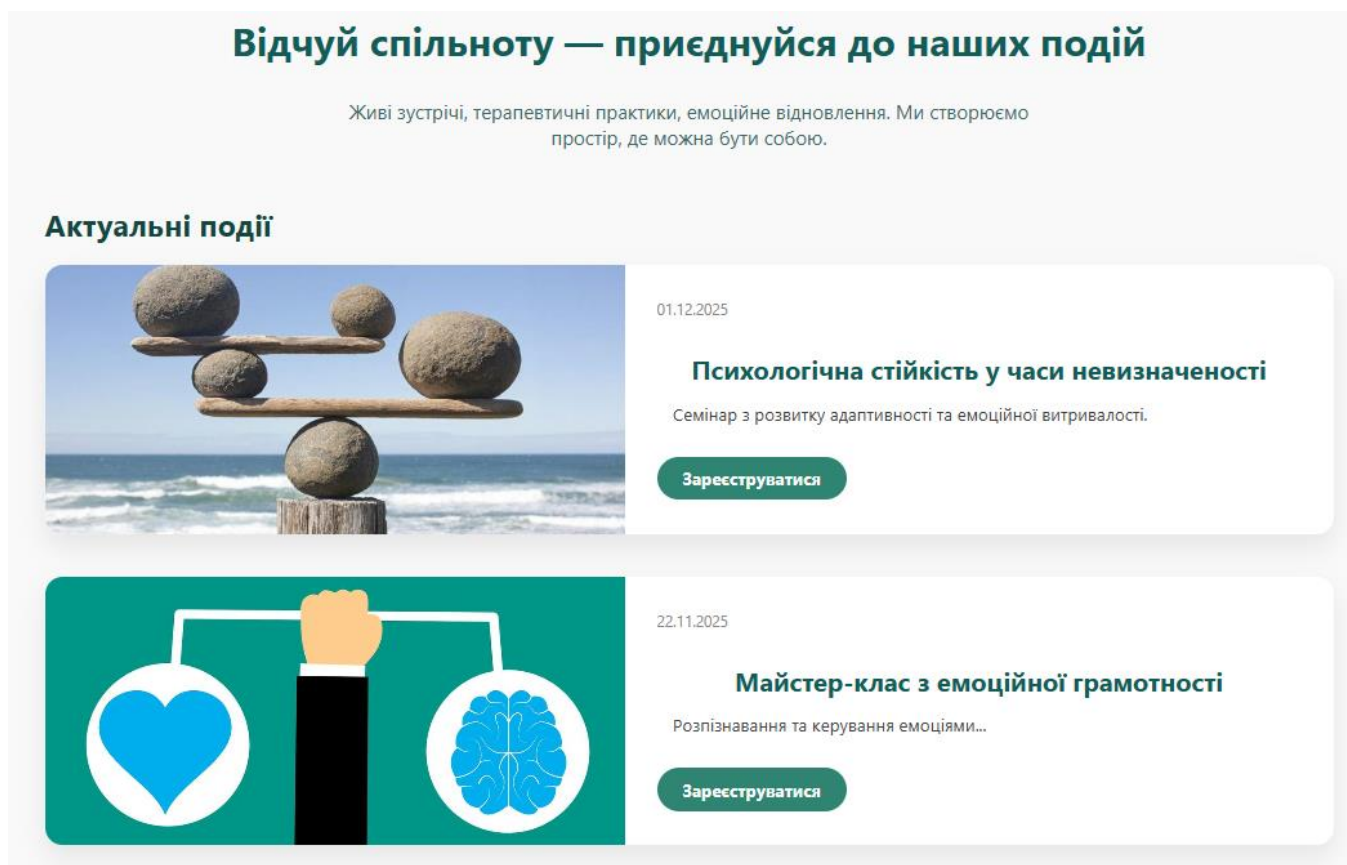


Рисунок 4.12 – Відображення актуальних подій

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Минулі заходи (рис. 4.13) подані окремо – вони свідчать про вже реалізовану активність платформи й формують у користувача довіру та інтерес. Завдяки якісним зображенням і чіткій структурі користувач швидко розуміє, про що була кожна подія.

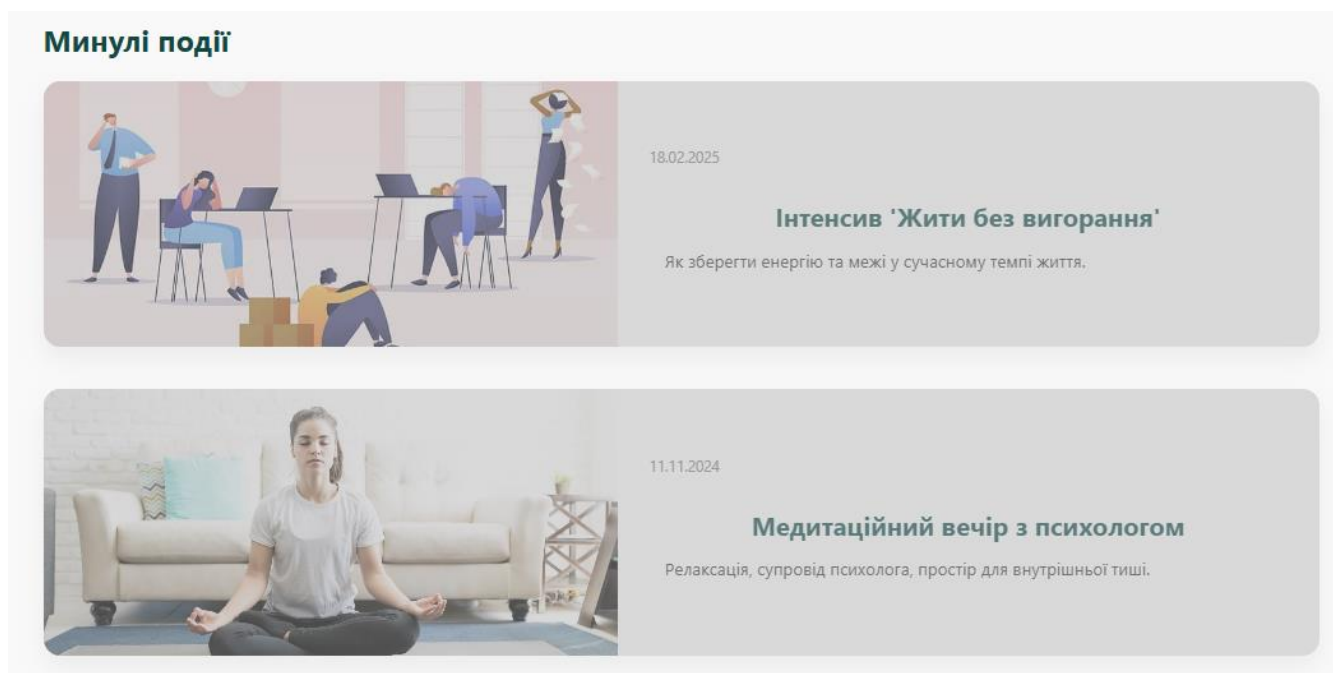


Рисунок 4.13 – Відображення минулих заходів

У нижній частині сторінки також інтегровано компонент Record, який пропонує пройти коротке опитування або обрати спеціаліста.

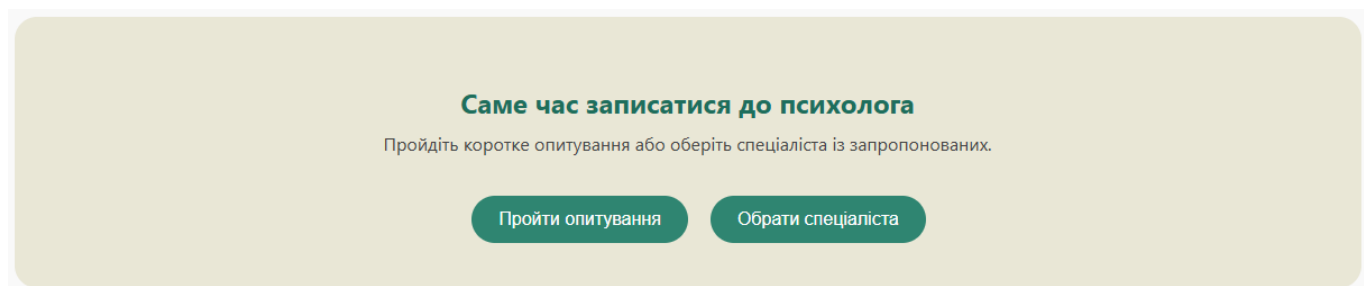


Рисунок 4.14 – Інтегрований компонент Record

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

Надалі користувач може перейти до сторінки «Блог» (рис. 4.15), що призначена для відображення публікацій з корисною інформацією або порадами для користувачів.

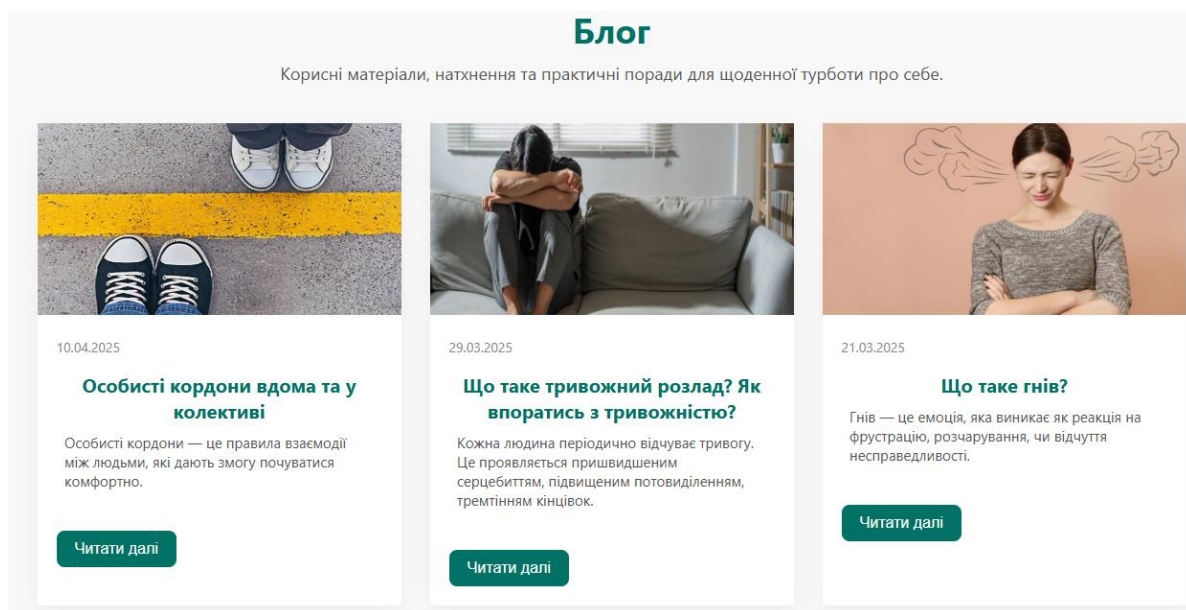


Рисунок 4.15 – Блок з відображенням публікацій

Вона є частиною клієнтської частини застосунку та має вигляд стрічки з тематичними записами, кожен з яких містить заголовок, дату публікації, зображення та короткий опис.

Для відображення кожного запису використовується окремий компонент `Record`, який інкапсулює в собі верстку одного елементу блогу. Компонент `Blog` імпортує масив об'єктів `posts`, що містить відповідні дані, і динамічно відображає їх за допомогою методу `map()`. Завдяки використанню `React`-компонентів та сучасного підходу до розробки `UI`, сторінка легко адаптується до змін вмісту або структури.

У нижній частині сторінки також інтегровано компонент `Record`, який пропонує пройти коротке опитування або обрати спеціаліста.

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

Сторінка «Контакти» реалізована як окремий React-компонент Contact і слугує для надання користувачам усієї необхідної інформації для зв'язку з організацією. Вона включає два основні елементи: блок із контактною інформацією та форму зворотного зв'язку (рис. 4.16).

Зв'яжіться з нами

Маєте запитання або хочете отримати консультацію? Ми завжди на зв'язку!

Контактна інформація

- Адреса: Київ, вул. Підтримки 15
- Телефон: +380 99 123 4567
- Email: dihajvilno@psyhelp.ua
- Графік: Пн-Пт 9:00 — 18:00

Напишіть нам

Іван Петренко

ivan.petrenko@gmail.com

Щиро дякую за зручний та інформативний сайт! Усе працює швидко. Було б чудово, якби додали темну тему або функцію нічного режиму. Успіхів у розвитку!

Надіслати повідомлення

Рисунок 4.16 – Блок із контактною інформацією та формою зворотного зв'язку

У верхній частині сторінки розміщено заголовок «Зв'яжіться з нами» та коротке пояснення призначення сторінки. Основний вміст поділено на дві колонки. Ліворуч міститься детальна контактна інформація, включно з адресою, телефоном, електронною поштою, графіком роботи, а також посиланнями на соціальні мережі: Instagram, Facebook і Telegram. Цей блок створює враження відкритості й доступності сервісу.

Праворуч знаходиться інтерактивна форма, яка дозволяє користувачам надіслати повідомлення безпосередньо зі сторінки. Вона містить поля для імені,

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

електронної пошти та тексту повідомлення. Усі поля є обов'язковими для заповнення, що дозволяє отримувати коректні та повноцінні запити. Після заповнення форми користувач має можливість натиснути кнопку «Надіслати повідомлення», яка ініціює процес відправлення даних.

Форма інтегрована з механізмом надсилання електронних листів на адресу Gmail, що забезпечує оперативне отримання повідомлень безпосередньо розробником або адміністратором вебресурсу.

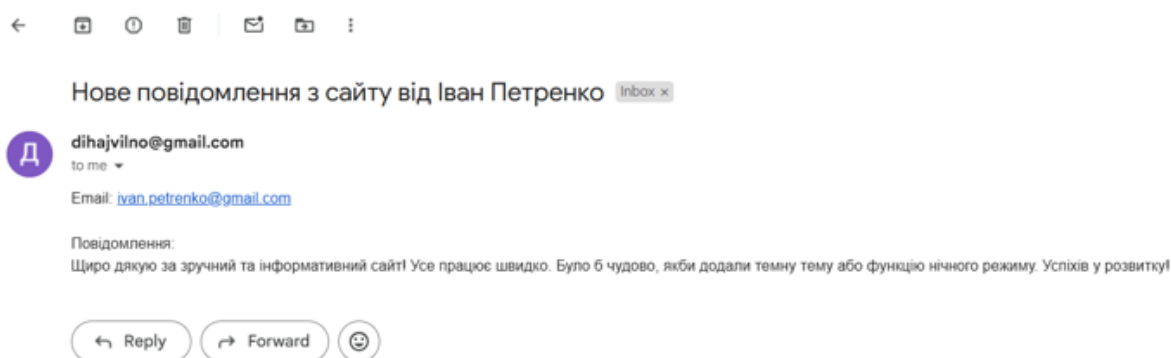


Рисунок 4.17 – Отримане повідомлення з вебформи у поштової скриньці Gmail

Таким чином, реалізовано простий, але ефективний канал зворотного зв'язку між користувачем та системою.

Після авторизації користувача у вебзастосунку йому стають доступні додаткові функціональні вкладки, що забезпечують швидкий доступ до персоналізованих розділів системи. Компонент UserMenu реалізує випадаюче меню, яке активується натисканням на аватар користувача у правому верхньому куті інтерфейсу. Меню автоматично закривається при кліку поза його межами, що підвищує зручність користування.

Залежно від ролі користувача (клієнт або психолог) у меню відображається відповідний набір пунктів. Таким чином, UserMenu відіграє ключову роль у навігації

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

та взаємодії користувача з системою, забезпечуючи доступ до чату, календаря, щоденника, налаштувань та інших функцій. Повний перелік маршрутів, які обробляють запити, пов'язані з користувачами, подано у Додатку Г.

В таблиці 4.1 наведено опис вкладок, їх призначення та доступність для різних типів користувачів.

Таблиця 4.1 – Навігаційні вкладки користувача та психолога

Назва вкладки	Опис	Доступно для
Чати	Інтерфейс для обміну повідомленнями з іншими користувачами.	Клієнт Психолог
Щоденник	Персональний простір клієнта для фіксації думок, емоцій.	Клієнт
Мої сесії	Відображає заплановані консультації, їх дати та статуси.	Клієнт
Календар сесій	Інтерфейс для перегляду, планування та редагування сесій із клієнтами.	Психолог
Анонімні запити	Секція з вхідними зверненнями від нових, ще неідентифікованих користувачів, що потребують допомоги.	Психолог
Налаштування	Сторінка редагування профілю: зміна особистих даних, паролю, фото тощо.	Клієнт Психолог
Вийти	Завершення сесії користувача, очищення авторизації та перенаправлення на головну сторінку застосунку.	Клієнт Психолог

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

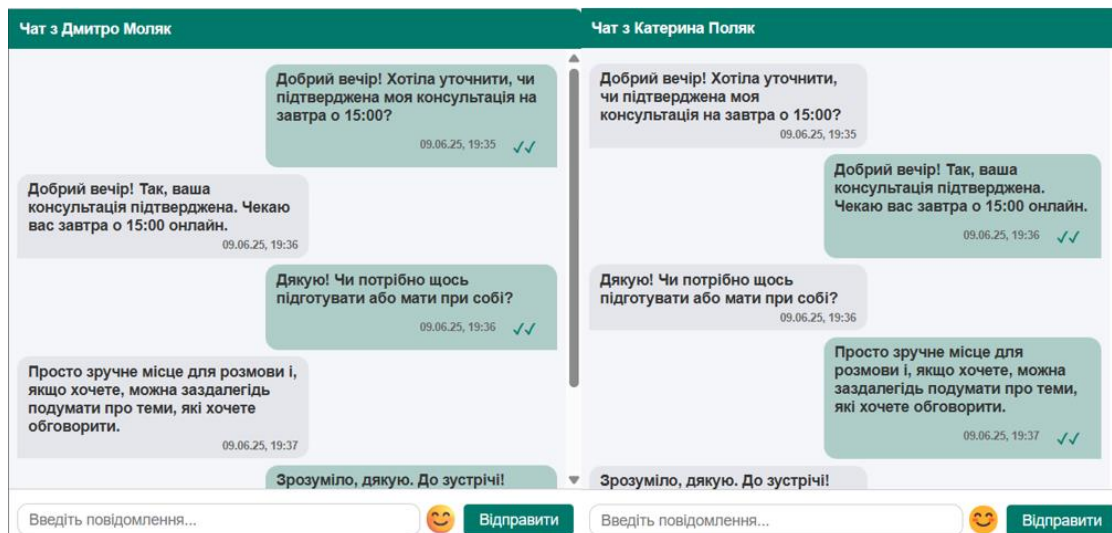


Рисунок 4.18 – Інтерфейс вкладки «Чати»

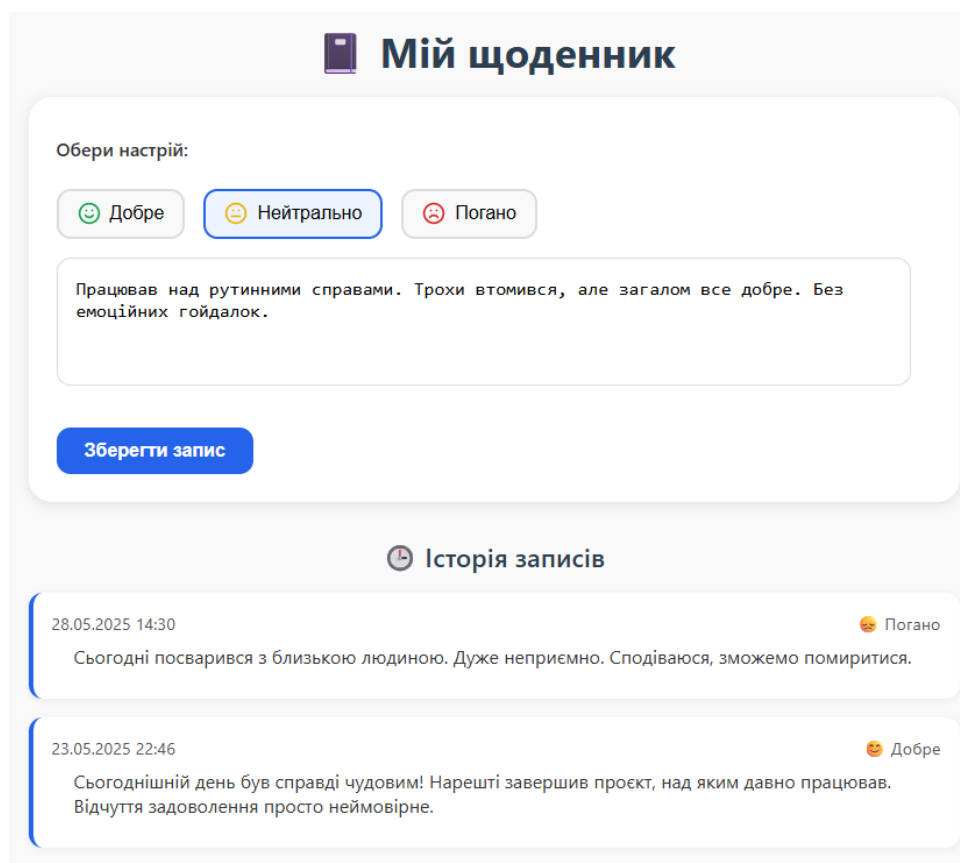


Рисунок 4.19 – Персональний щоденник користувача

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

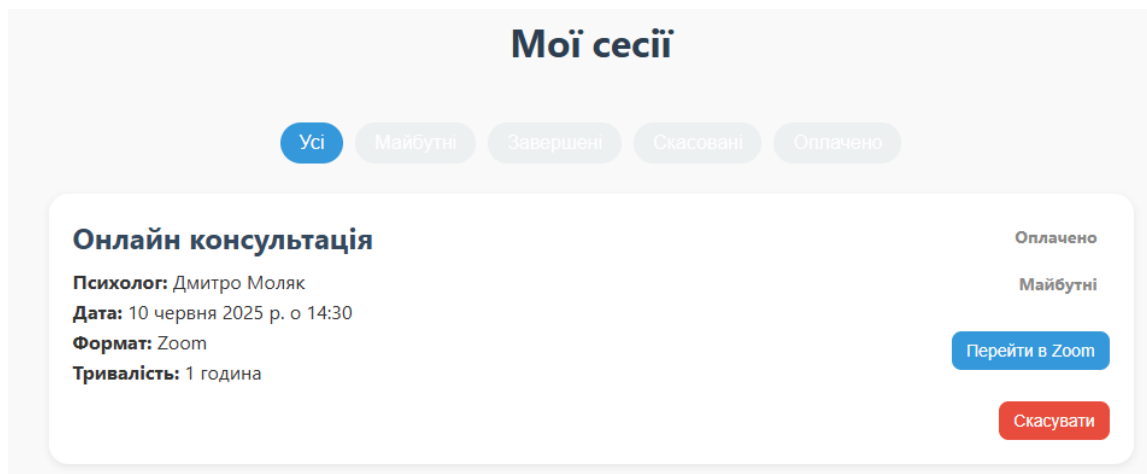


Рисунок 4.20 – Сторінка «Мої сесії» з інформацією про заплановану консультацію

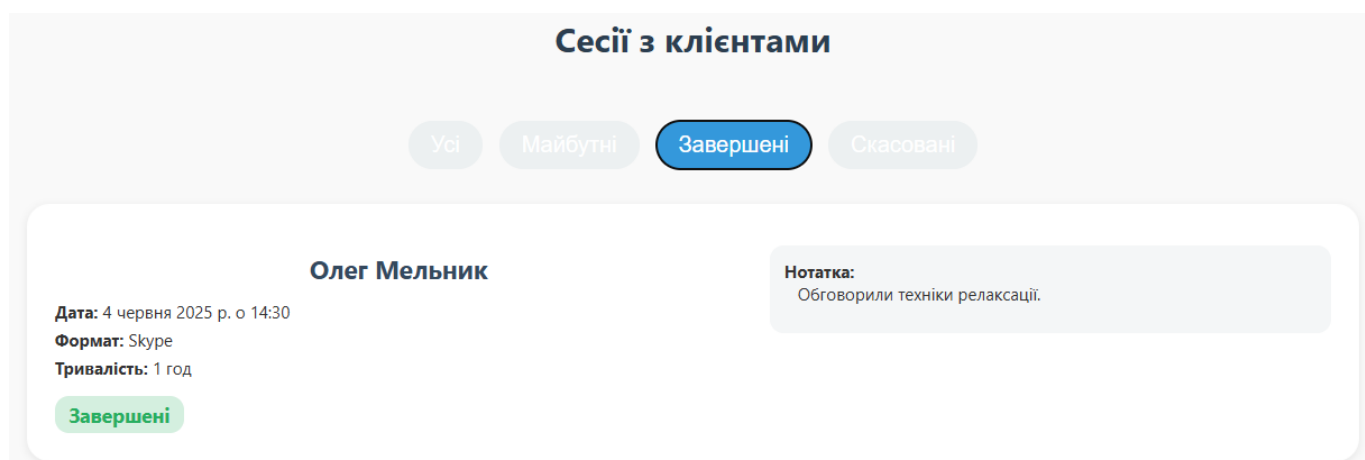


Рисунок 4.21 – Календар сесій для психолога

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

Анонімні звернення

Доброго дня. Мені складно говорити про це з кимось знайомим, тому вирішила звернутись анонімно. Оста...
04.06.2025, 12:27

Анонімна підтримка
Завершити сесію

Якщо ви випадково вийшли з чату — просто поверніться за цим посиланням.
Ваш сеанс зберігається протягом обмеженого часу.

Анонім: Доброго дня. Мені складно говорити про це з кимось знайомим, тому вирішила звернутись анонімно. Останнім часом я відчуваю сильне емоційне виснаження, хоча зовні все виглядає нормально. Мені важко прокидатися вранці, нічого не хочеться, і я часто відчуваю провину, ніби нічого не роблю достатньо добре. Я не впевнена, що це депресія, але починаю хвилюватись. Просто хотілося б зрозуміти, що зі мною і чи можна якось собі допомогти. Дякую, що прочитали.

Психолог: Дякую, що поділилися. Ви вже зробили важливий крок, звернувшись. Те, що ви описуєте — емоційне виснаження, апатію, почуття провини — справді можуть бути ознаками психоемоційного перенапруження або навіть початкової стадії депресії. Чи було щось останнім часом, що сильно змінило ваше життя або викликало додатковий стрес?

Анонім: Так, у мене була складна ситуація на роботі, плюс я майже не відпочивав(-ла) останні місяці. Постійно здається, що потрібно працювати ще більше, щоб не «випасти з ритму».

Психолог: Це дуже знайоме багатьом. Ваш організм і психіка сигналізують, що їм потрібен перепочинок. Перший крок — це визнати свої відчуття і дозволити собі відпочивати без провини. Ми можемо разом розробити стратегії відновлення і поговорити про баланс між роботою і особистим життям.

Надіслати

Рисунок 4.22 – Перегляд анонімних запитів та приклад використання чату

Ім'я
Дмитро

Прізвище
Моляк

Дата народження
19.08.1987

Електронна пошта
dmolyak@gmail.com

Пароль

Підтвердження пароля

Зберегти

Вибрати файл

Файл не вибрано

Рисунок 4.23 – Сторінка налаштувань профілю

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Наведені вище ілюстрації відображають зовнішній вигляд кожної вкладки застосунку та демонструють основні функціональні елементи інтерфейсу. Для кращого розуміння логіки роботи клієнтської частини вебзастосунку, у Додатку Б наведено відповідну карту інтерфейсу.

4.3 Тестування ПЗ

Значна частина часу та ресурсів, витрачених на розробку програмного забезпечення, припадає на його тестування — критично важливий етап, що охоплює комплекс заходів з перевірки, верифікації та валідації програмного продукту. Тестування програмного забезпечення є систематичним процесом вивчення та аналізу функціональності, надійності, безпеки, продуктивності та сумісності програми з метою виявлення дефектів, логічних помилок, невідповідностей технічним вимогам або очікуванням користувача. Цей процес не лише спрямований на пошук і усунення помилок, але й на підтвердження того, що програмний продукт повною мірою виконує свою передбачену функцію, відповідає встановленим стандартам якості, є стабільним у роботі та готовим до використання кінцевими споживачами. Таким чином, тестування виступає ключовим інструментом забезпечення якості (QA) і є невід’ємною частиною життєвого циклу розробки програмного забезпечення.

Для перевірки коректності результату розробки ПЗ «Вебзастосунок для служби психологічної підтримки осіб» розроблено деякі тестові сценарії, які наведено в табл. 4.2 – 4.8.

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Таблиця 4.2 – Сценарій тестування «Реєстрація»

ID	1
Мета	Створити обліковий запис
Передумови	1) користувач відкрив сторінку реєстрації; 2) користувач не авторизований у вебзастосунку; 3) користувач не має облікового запису.
Успішний сценарій	1) користувач заповнює обов'язкові поля: ім'я, прізвище, пошту, пароль та підтвердження паролю; 2) користувач натискає кнопку «Зареєструватися». 3) система виконує валідацію даних; 4) система відкриває користувачу сторінку його профілю.
Розширення	1) користувач не заповнює обов'язкові поля; 2) система виводить повідомлення про необхідність заповнення обов'язкових полів.
	1) користувач не заповнює обов'язкові поля; 2) система виводить повідомлення про необхідність заповнення обов'язкових полів.
	1) пароль та підтвердження паролю не збігаються; 2) система виводить повідомлення відповідне повідомлення.
Статус	Виконано

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Таблиця 4.3 – Сценарій тестування «Авторизація»

ID	2
Мета	Авторизуватися у вебзастосунку
Передумови	1) користувач відкрив сторінку авторизації; 2) користувач вже зареєстрований.
Успішний сценарій	1) користувач вводить пошту та пароль у відповідні поля. 2) користувач натискає кнопку «Увійти»; 3) система відкриває користувачу сторінку його профілю.
Розширення	1) користувач вводить некоректні/неіснуючі у БД дані (email або пароль); 2) система виводить повідомлення про неправильно введені дані.
	1) користувач не заповнює обов'язкові поля; 2) система виводить повідомлення про необхідність заповнення таких полів.
Статус	Виконано

Таблиця 4.4 – Сценарій тестування «Редагування профілю користувача»

ID	3
Мета	Редагувати профіль
Передумови	Користувач авторизований у системі
Успішний сценарій	1) користувач переходить на сторінку редагування профілю; 2) користувач завантажує фото, додає опис, змінює ім'я та прізвище; 3) користувач натискає кнопку «Зберегти»; 4) система зберігає внесені користувачем зміни.
Розширення	1) користувач натискає на кнопку «Змінити пароль»; 2) система відкриває користувачу форму зміни паролю; 3) користувач вводить новий пароль та підтверджує його; 4) система зберігає новий пароль.
Статус	Виконано

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Таблиця 4.5 – Сценарій тестування «Запис на консультацію»

ID	4
Мета	Запис користувача на консультацію
Передумови	Користувач авторизований у системі
Успішний сценарій	1) користувач переходить на сторінку запису на консультацію; 2) вибирає дату, час та спеціаліста; 3) натискає кнопку «Оплатити»; 4) система перенаправляє користувача на платіжну сторінку LiqPay, де він виконує оплату; 5) після підтвердження оплати система зберігає інформацію в базі даних.
Розширення	1) якщо обрана дата зайнята – система пропонує альтернативні варіанти; 2) користувач отримує повідомлення про успішний запис.
Статус	Виконано

Таблиця 4.6 – Сценарій тестування «Перегляд та відповідь на анонімні запити»

ID	5
Мета	Переглянути запити та надати відповідь
Передумови	Користувач авторизований у системі як психолог
Успішний сценарій	1) психолог переходить до розділу анонімних запитів; 2) вибирає запит зі списку; 3) читає повідомлення та вводить відповідь; 4) натискає кнопку «Надіслати»; 5) система зберігає відповідь та прикріплює її до запиту.
Розширення	1) якщо запит вже закрито – система показує повідомлення про це; 2) консультант може прикріпити файл або посилання у відповіді.
Статус	Виконано

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Таблиця 4.7 – Сценарій тестування «Проходження психологічного тесту»

ID	6
Мета	Пройти психологічний тест
Передумови	Користувач авторизований у системі
Успішний сценарій	1) користувач обирає тест; 2) проходить послідовно усі питання, обираючи відповіді; 3) натискає кнопку «Завершити тест»; 4) система обробляє результати та виводить інтерпретацію.
Розширення	1) якщо користувач не відповів на всі питання – система просить на них відповіді; 2) результати зберігаються у профілі користувача.
Статус	Виконано

Таблиця 4.8 – Сценарій тестування «Зміна налаштувань даних користувача»

ID	7
Мета	Змінити особисті дані користувача
Передумови	Користувач авторизований у системі
Успішний сценарій	1) користувач переходить до розділу «Налаштування»; 2) змінює доступні параметри; 3) натискає кнопку «Зберегти»; 4) система зберігає внесені зміни; 5) інтерфейс оновлюється відповідно до нових налаштувань.
Розширення	1) при збереженні налаштувань виникає помилка – система повідомляє користувача про це; 2) налаштування зберігаються локально або на сервері й застосовуються після повторного входу.
Статус	Виконано

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Висновки до розділу 4

У четвертому розділі кваліфікаційної роботи наведено аспекти програмної реалізації вебзастосунку. Наведено частини розробленого програмного коду для деяких функцій системи та сторінок, зокрема головної сторінки вебзастосунку. Також наведено список деяких моделей, що відповідають за функції системи.

Розроблено керівництво користувача: наведено приклади сторінок, до яких має доступ звичайний користувач, описано деякий функціонал та сторінки, по яким може перейти користувач.

Проведено тестування розробленого ПЗ. Для цього, розроблено тестові сценарії, що описують як успішний сценарій, так і можливі розширення.

При виконанні тестових сценаріїв для розробленого ПЗ виявлено деякі недоліки та помилки у проєкті, після чого їх виправлено. Описані сценарії тестування успішно виконано.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи реалізовано повнофункціональний вебзастосунок, призначений для надання психологічної підтримки в онлайн-форматі. Розроблена система враховує актуальні соціальні виклики, зумовлені війною в Україні, та забезпечує зручний і безпечний інструмент для взаємодії між користувачами та фахівцями-психологами.

У межах проєкту було виконано наступні завдання:

- проаналізовано предметну область та існуючі сервіси психологічної підтримки;
- обґрунтовано вибір технологічного стеку (React.js, Node.js, Express, MongoDB, JWT);
- спроектовано архітектуру клієнтської та серверної частин застосунку;
- реалізовано ключові функціональні модулі: реєстрацію, авторизацію, особистий кабінет, чат у реальному часі, запис на консультацію, модуль анонімного звернення, блог із порадами та тестуванням психоемоційного стану;
- забезпечено безпеку даних користувачів відповідно до сучасних стандартів (використання bcrypt, JWT).

Результатом роботи стала сучасна, адаптивна та масштабована вебплатформа з україномовним інтерфейсом, яка надає можливість отримати психологічну підтримку швидко, анонімно та зручно.

Проведене тестування вебзастосунку показало його працездатність і відповідність поставленим функціональним вимогам. Реалізовані модулі успішно інтегруються між собою, забезпечуючи цілісність системи та позитивний користувацький досвід.

У майбутньому планується розширення можливостей застосунку за рахунок додавання відеоконференцій, мобільного додатка, інструментів самодопомоги в

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

офлайн-режимі, а також штучного інтелекту для попереднього аналізу психологічного стану користувача.

Таким чином, розроблений вебзастосунок є вагомим внеском у цифрову сферу психологічної підтримки та має потенціал до практичного використання в умовах воєнного та післявоєнного періоду.

З вихідним кодом демонстраційного проєкту можна ознайомитися на GitHub за посиланням: <https://github.com/nadik-k/Psychology-Site>.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Всесвітня організація охорони здоров'я. Психічне здоров'я: укріплення відповіді // WHO. 2022. URL: <https://www.who.int/news-room/fact-sheets/detail/mental-health-strengthening-our-response> (дата звернення 26.04.2025).
2. BetterHelp. Онлайн-консультування з ліцензованими психологами. URL: <https://www.betterhelp.com> (дата звернення 28.04.2025).
3. Talkspace. Онлайн-терапія з професійними психологами. URL: <https://www.talkspace.com> (дата звернення 28.04.2025).
4. «Ти як?». Платформа психологічної підтримки в кризових ситуаціях. URL: <https://howareu.com/> (дата звернення 28.04.2025).
5. «Розкажи мені». Безкоштовна психологічна допомога українцям. URL: <https://tellme.com.ua/> (дата звернення 28.04.2025).
6. Mindly. Додаток для підтримки ментального здоров'я. URL: <https://mindlyspace.com/> (дата звернення 28.04.2025).
7. Гончаренко А. В. Вебзастосунок для психологічної підтримки під час війни. Кваліфікаційна робота. ЧНУ ім. П. Могили. 2023. URL: <https://krs.chmnu.edu.ua/jspui/bitstream/123456789/2958/1/%D0%93%D0%BE%D0%BD%D1%87%D0%B0%D1%80%D0%B5%D0%BD%D0%BA%D0%BE%20%D0%90%D0%BB%D1%96%D0%BD%D0%B0%20%D0%91%D0%9A%D0%A0.pdf> (дата звернення: 01.05.2025).
8. П'явчук П. С. Розробка програмного засобу для щоденної психологічної підтримки користувача мовою Python. Кваліфікаційна робота. ДУІКТ. 2024. URL: <https://duikt.edu.ua/repozitorii/ipz/2024/%D0%9F%D0%94-44/%D0%9F%D0%94-44%20%D0%9F%D1%8F%D0%B2%D1%87%D1%83%D0%BA%20%D0%9F.%D0%A1..pdf> (дата звернення: 01.05.2025).
9. Шпирналь І. М. Розробка веб-платформи для ведення блогу психологічної

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

підтримки. Кваліфікаційна робота. УКД. 2024. URL:
<http://repository.ukd.edu.ua/bitstream/handle/123456789/840/%d0%a8%d0%bf%d0%b8%d1%80%d0%bd%d0%b0%d0%bb%d1%8c%20%d0%86%d1%80%d0%b8%d0%bd%d0%b0.pdf?sequence=1&isAllowed=y> (дата звернення: 01.05.2025)

10. Draw.io an online diagramming and flowcharting tool. URL: <http://draw.io> (дата звернення: 10.05.2025).

11. REST API Tutorial – What is REST? URL: <https://restfulapi.net> (дата звернення: 11.05.2025).

12. Bootstrap Documentation – Build fast, responsive sites with Bootstrap. URL: <https://getbootstrap.com> (дата звернення: 11.05.2025).

13. MDN Web Docs – Resources for Developers. URL: <https://developer.mozilla.org> (дата звернення: 11.05.2025).

14. React – The library for web and native user interfaces. URL: <https://react.dev> (дата звернення: 11.05.2025).

15. Express.js – Fast, unopinionated, minimalist web framework for Node.js. URL: <https://expressjs.com> (дата звернення: 11.05.2025).

16. Node.js v24.2.0 documentation. URL: <https://nodejs.org/en/docs> (дата звернення: 11.05.2025).

17. MongoDB Documentation – The official MongoDB Documentation. URL: <https://www.mongodb.com/docs> (дата звернення: 12.05.2025).

18. JWT – Introduction to JSON Web Tokens. URL: <https://jwt.io/introduction> (дата звернення: 15.05.2025).

19. Mongoose Documentation – MongoDB object modeling tool designed to work in an asynchronous environment. URL: <https://mongoosejs.com/docs/> (дата звернення: 18.05.2025).

20. Postman documentation – The place to find official information on how to use

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

Postman in your API projects. URL: <https://learning.postman.com/docs/getting-started/introduction/> (дата звернення: 18.05.2025).

21. Socket.IO Documentation – Bidirectional and low-latency communication for every platform. URL: <https://socket.io/docs/v4/> (дата звернення: 18.05.2025).

22. NPM Docs – Documentation for the npm registry, website, and command-line interface. URL: <https://docs.npmjs.com/> (дата звернення: 18.05.2025).

23. VSCode Your code editor. Redefined with AI. URL: <https://code.visualstudio.com> (дата звернення 18.05.2025).

24. Dayley, B. (2014). Node. js, MongoDB, and AngularJS web development. Addison-Wesley Professional.

25. Vipul, A. M., & Sonpatki, P. (2016). ReactJS by Example-Building Modern Web Applications with React. Packt Publishing Ltd.

26. Bradshaw, S., Brazil, E., & Chodorow, K. (2019). MongoDB: the definitive guide: powerful and scalable data storage. " O'Reilly Media, Inc."

27. Patil, M. M., Hanni, A., Tejeshwar, C. H., & Patil, P. (2017, February). A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retrieval operations using a web/android application to explore load balancing—Sharding in MongoDB and its advantages. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 325-330). IEEE.

28. Shingala, K. (2019). Json web token (jwt) based client authentication in message queuing telemetry transport (mqtt). arXiv preprint arXiv:1903.02895.

29. Jones, M., Campbell, B., & Mortimore, C. (2015). Json web token (jwt) profile for oauth 2.0 client authentication and authorization grants (No. rfc7523).

30. Holmes, S. (2013). Mongoose for Application Development. Packt Publishing Ltd.

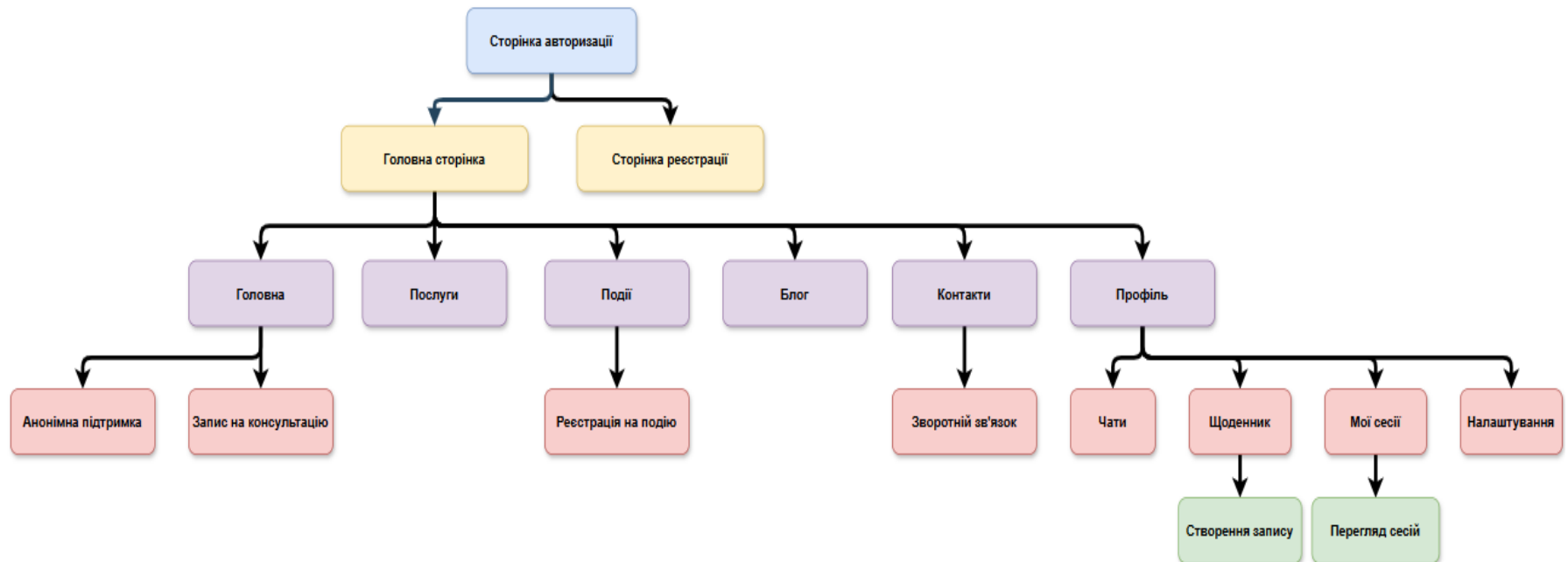
ДОДАТОК А

Діаграма варіантів використання для акторів



Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

ДОДАТОК Б
Карта вебзастосунку для клієнта



Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

ДОДАТОК В

Структура серверної частини проєкту

```
// server/server.js
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
const path = require('path');
const http = require('http');
const { Server } = require('socket.io');
const JWT_SECRET = 'superSecretKey123';
const jwt = require('jsonwebtoken');
const bodyParser = require('body-parser');

const userRoutes = require('./routes/userRoutes');
const cardRoutes = require('./routes/cardRoutes');
const diaryRoutes = require('./routes/diaryRoutes');
const contactRoute = require('./routes/contact');
const messageRoutes = require('./routes/messageRoutes');
const anonymousRoutes = require('./routes/anonymousRoutes');
const Message = require('./models/messageModel');
const AnonymousMessage = require('./models/anonymousModel');
const postRoutes = require('./routes/postRoutes');
const eventRoutes = require('./routes/eventRoutes');
const statisticsRouter = require('./routes/statisticRoutes');
const liqpayRoutes = require('./routes/liqpayRoutes');
const sessionRoutes = require('./routes/sessionRoutes');

const app = express();
const server = http.createServer(app);
const io = new Server(server, {
  cors: {
    origin: 'http://localhost:3000',
    credentials: true,
  }
});

app.use(cors({
  origin: 'http://localhost:3000',
  credentials: true
}));
app.use(express.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

mongoose.connect('mongodb+srv://nkudrya2004:VdZCvxBBNywppNrY@cluster0.81czigz.mongodb.net/Breathe_freely?retryWrites=true&w=majority')
  .then(() => console.log('Connected to MongoDB'))
  .catch((err) => console.error('MongoDB error:', err));

app.use('/api/users', userRoutes);
app.use('/api/cards', cardRoutes);
```

Кафедра інтелектуальних інформаційних систем

Вебзастосунок для служби психологічної підтримки осіб

```

app.use('/uploads', express.static(path.join(__dirname, 'uploads')));
app.use('/api/diary', diaryRoutes);
app.use('/api', contactRoute);
app.use('/api/messages', messageRoutes);
app.use('/api/anonymous', anonymousRoutes);
app.use('/api/posts', postRoutes);
app.use('/api/events', eventRoutes);
app.use('/api/static', statisticsRouter);
app.use('/api', liqpayRoutes);
app.use('/api/sessions', sessionRoutes);

io.on('connection', (socket) => {
  console.log('Користувач підключився:', socket.id);
  socket.on('send_message', async (data) => {
    try {
      const savedMsg = new Message(data);
      await savedMsg.save();
      io.emit('receive_message', savedMsg);
    } catch (err) {
      console.error('Помилка при збереженні повідомлення:', err);
    }
  });

  socket.on('read_messages', async ({ readerId, senderId }) => {
    try {
      await Message.updateMany(
        { senderId, receiverId: readerId, read: false },
        { $set: { read: true } }
      );
      io.emit('messages_read', { readerId, senderId });
    } catch (err) {
      console.error('Помилка при оновленні статусу прочитання:', err);
    }
  });

  socket.on('receive_message', (data) => {
    if (data.senderId !== myId) {
      setMessages(prev => [...prev, data]);

      if (activeUser && activeUser._id === data.senderId) {
        socket.emit('read_messages', {
          readerId: myId,
          senderId: data.senderId
        });
      }
    }
  });

  socket.on('create_anonymous_session', () => {
    const sessionId = uuidv4();
    anonymousChats[sessionId] = [];
    socket.join(sessionId);
    socket.emit('anonymous_session_created', { sessionId });
  });

  socket.on('join_anonymous_session', ({ sessionId }) => {

```

Кафедра інтелектуальних інформаційних систем

Вебзастосунок для служби психологічної підтримки осіб

```

    if (anonymousChats[sessionId]) {
      socket.join(sessionId);
      socket.emit('anonymous_session_joined', { sessionId });
    } else {
      socket.emit('error', { message: 'Сесію не знайдено.' });
    }
  });

  socket.on('anon_join', (sessionId) => {
    socket.join(sessionId);
    console.log(`Користувач ${socket.id} приєднався до сесії: ${sessionId}`);
  });

  socket.on('anon_send_message', async (data) => {
    const { sessionId, text, authToken } = data;

    if (!sessionId || !text) {
      console.error('Недостатньо даних для надсилання повідомлення');
      return;
    }

    let sender = 'anonymous';
    if (authToken) {
      try {
        const decoded = jwt.verify(authToken, JWT_SECRET);
        if (decoded.role === 'psycholog') {
          sender = 'psycholog';
        }
      } catch (err) {
        console.error('Невірний токен:', err);
      }
    }

    const newMessage = new AnonymousMessage({ sessionId, text, sender });
    try {
      await newMessage.save();
      io.to(sessionId).emit('anon_receive_message', newMessage);
    } catch (err) {
      console.error('Помилка збереження повідомлення:', err);
    }
  });
});

server.listen(5000, () => {
  console.log('Сервер запущено на http://localhost:5000');
});

// server/routes/sessionRoutes.js

```

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

ДОДАТОК Г

Маршрути для обробки запитів, пов'язаних з користувачами

```
const express = require('express');
const router = express.Router();
const User = require('../models/User');
const { JWT_SECRET } = require('../config');
const authenticateToken = require('../middleware/auth');
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken');
const multer = require('multer');
const path = require('path');
const Psychologist = require('../models/Card');

const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, 'uploads/');
  },
  filename: function (req, file, cb) {
    const uniqueSuffix = Date.now() + '-' + Math.round(Math.random() * 1E9);
    const ext = path.extname(file.originalname);
    cb(null, file.fieldname + '-' + uniqueSuffix + ext);
  }
});

const upload = multer({ storage });

router.get('/profile', authenticateToken, async (req, res) => {
  const user = await User.findById(req.user.id).select('-password');
  res.json(user);
});

router.get('/', async (req, res) => {
  try {
    const users = await User.find();
    res.json(users);
  } catch (err) {
    res.status(500).json({ error: 'Server error' });
  }
});

router.post('/', async (req, res) => {
  try {
    const newUser = new User(req.body);
    await newUser.save();
    res.status(201).json(newUser);
  } catch (err) {
    res.status(400).json({ error: err.message });
  }
});

function isValidEmail(email) {
  const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  return re.test(email);
}
```

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

```

}

function isValidPassword(password) {
  return password.length >= 6;
}

router.post('/register', async (req, res) => {
  const {
    firstName,
    lastName,
    gender,
    birthdate,
    email,
    password,
    role,
    photo,
    experience,
    topics,
    price
  } = req.body;

  if (!isValidEmail(email)) {
    return res.status(400).json({ message: 'Невірний формат email' });
  }

  if (!isValidPassword(password)) {
    return res.status(400).json({ message: 'Пароль повинен містити щонайменше 6 символів' });
  }

  try {
    const existingUser = await User.findOne({ email });
    if (existingUser) {
      return res.status(409).json({ message: 'Користувач з таким email вже існує' });
    }
  }

  const hashedPassword = await bcrypt.hash(password, 10);
  const newUser = new User({
    firstName,
    lastName,
    gender,
    birthdate,
    email,
    password: hashedPassword,
    role,
    photo: photo ? photo : undefined,
    createdAt: new Date().toISOString(),
    experience: role === 'psycholog' ? experience : undefined,
    topics: role === 'psycholog' ? topics : [],
    price: role === 'psycholog' ? price : undefined
  });
  const name = firstName + " " + lastName
  if (role === 'psycholog') {
    const newPsychologist = new Psychologist({
      userId: newUser._id,

```

Кафедра інтелектуальних інформаційних систем

Вебзастосунок для служби психологічної підтримки осіб

```

        name: name,
        experience,
        photo,
        topics,
        price
    });
    console.log("It's good!")
    await newPsychologist.save();
}
await newUser.save();

const authToken = jwt.sign(
    { id: newUser._id, email: newUser.email, role: newUser.role },
    JWT_SECRET,
    { expiresIn: '7d' }
);

res.status(201).json({ authToken });
} catch (error) {
    console.error('Registration error:', error);
    res.status(500).json({ message: 'Помилка сервера під час реєстрації' });
}
});

router.post('/login', async (req, res) => {
    const { email, password } = req.body;

    try {
        const user = await User.findOne({ email });
        if (!user) {
            return res.status(404).json({ message: 'Користувача не знайдено' });
        }

        const isMatch = await bcrypt.compare(password, user.password);
        if (!isMatch) {
            return res.status(401).json({ message: 'Невірний пароль' });
        }

        const token = jwt.sign(
            { id: user._id, email: user.email, role: user.role, photo: user.photo },
            JWT_SECRET,
            { expiresIn: '7d' }
        );

        res.json({
            token,
            user: {
                id: user._id,
                email: user.email,
                role: user.role,
                photo: user.photo,
            }
        });
    } catch (error) {
        console.error('Login error:', error);
        res.status(500).json({ message: 'Помилка сервера при вході' });
    }
});

```

Кафедра інтелектуальних інформаційних систем Вебзастосунок для служби психологічної підтримки осіб

```

    }
  });

  router.get('/:id', async (req, res) => {
    try {
      const user = await User.findById(req.params.id).select('-password');
      if (!user) return res.status(404).json({ message: 'Користувача не знайдено' });
      res.json(user);
    } catch (err) {
      console.error('Помилка отримання користувача:', err);
      res.status(500).json({ message: 'Помилка сервера' });
    }
  });

  router.put('/update', authenticateToken, upload.single('photo'), async (req, res) => {
    try {
      const userId = req.user.id;
      const {
        firstName,
        lastName,
        birthdate,
        email,
        password,
      } = req.body;

      const name = firstName + " " + lastName
      const user = await User.findById(userId);
      if (!user) return res.status(404).json({ message: 'Користувача не знайдено' });

      if (email && email !== user.email) {
        const emailExists = await User.findOne({ email: email });
        if (emailExists) {
          return res.status(400).json({ message: 'Цей email вже зайнятий іншим користувачем' });
        }
        user.email = email;
      }

      if (firstName) user.firstName = firstName;
      if (lastName) user.lastName = lastName;
      if (birthdate) user.birthdate = birthdate;

      if (req.file) {
        user.photo = `/uploads/${req.file.filename}`;
      }

      if (password) {
        const hashedPassword = await bcrypt.hash(password, 10);
        user.password = hashedPassword;
      }

      await Psychologist.updateOne({ name: name }, { photo: user.photo });
      await user.save();
    }
  });

```

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

```
const updatedToken = jwt.sign(
  {
    id: user._id,
    email: user.email,
    role: user.role,
    photo: user.photo
  },
  JWT_SECRET,
  { expiresIn: '7d' }
);

res.json({
  message: 'Профіль оновлено',
  user,
  token: updatedToken
});
} catch (err) {
  console.error('Update error:', err);
  res.status(500).json({ message: 'Помилка при оновленні профілю' });
}
});

router.get('/chat-partners/:userId', async (req, res) => {
  const user = await User.findById(req.params.userId);
  if (!user) return res.status(404).json({ message: 'User not found' });

  const oppositeRole = user.role === 'client' ? 'psychologist' : 'client';
  const users = await User.find({ role: oppositeRole });
  res.json(users);
});

module.exports = router;
```


Кафедра інтелектуальних інформаційних систем
Вебзастосунок для служби психологічної підтримки осіб

ДОДАТОК Д

Реалізація інтеграції LiqPay для онлайн-оплат консультацій

```
// server/liqpay.js
const crypto = require('crypto');
require("dotenv").config({ path: "./config.env" })

function base64(str) {
  return Buffer.from(str).toString('base64');
}
function sha1(str) {
  return crypto.createHash('sha1').update(str).digest();
}
function signature(str) {
  return base64(sha1(str));
}
function generateDataAndSignature(params) {
  const fullParams = {
    ...params,
    public_key
  };
  const json = JSON.stringify(fullParams);
  const data = base64(json);
  const sign = signature(private_key + data + private_key);
  return { data, signature: sign };
}
module.exports = { generateDataAndSignature };

// server/routes/liqpayRoutes.js
const express = require('express');
const router = express.Router();
const { generateDataAndSignature } = require('../liqpay');
const Consultation = require('../models/Consultation');
const { Buffer } = require('buffer');

router.post('/pay', (req, res) => {
  const { amount, description } = req.body;

  const orderId = `order-${Date.now()}`;
  const { data, signature } = generateDataAndSignature({
    amount,
    currency: 'UAH',
    description,
    order_id: orderId,
    action: 'pay',
    version: '3',
    language: 'uk',
    result_url: 'http://localhost:3000/meetings',
    server_url: 'https://21f2-62-16-15-253.ngrok-free.app/api/callback',
    sandbox: 1
  });

  res.json({ data, signature });
});
```

Кафедра інтелектуальних інформаційних систем

Вебзастосунок для служби психологічної підтримки осіб

```
});

router.post('/callback', async (req, res) => {
  try {
    const { data, signature } = req.body;
    console.log('CALLBACK від LiqPay:', { data, signature });

    const json = JSON.parse(Buffer.from(data, 'base64').toString('utf-8'));
    console.log('LiqPay decoded data:', json);

    if (json.status === 'success' || json.status === 'sandbox') {
      const customInfo = JSON.parse(json.description.match(/{.*}/)?.[0] ||
'{}');
      const { userId, psychologistId, datetime, format } = customInfo;

      await Consultation.create({
        userId,
        psychologistId,
        datetime,
        format,
        status: 'paid',
        psychologistStatus: "Очікуйте"
      });

      console.log('Консультацію збережено');
    }

    res.sendStatus(200);
  } catch (err) {
    console.error('Помилка при обробці callback:', err);
    res.status(500).send('Server Error');
  }
});

module.exports = router;
```