

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»**

**РОЗРОБКА ТА ДОСЛІДЖЕННЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ СТЕЖЕННЯ
ЗА МЕНТАЛЬНИМ ЗДОРОВ'ЯМ**

**DEVELOPMENT AND RESEARCH OF A WEB APPLICATION FOR
MONITORING MENTAL HEALTH**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗм-21
Гергель В. В.
Керівник:
д.т.н., професор
Андрущак І. Є.

Кваліфікаційну роботу
допущено до захисту
«__» _____ 20__ р.
Гарант освітньої програми:
к.т.н., доцент Суринович О. М.

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення
Ступінь вищої освіти магістр
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 121 «Інженерія програмного забезпечення»
Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри

«__» _____ 202__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ДРУГОГО (МАГІСТЕРСЬКОГО) РІВНЯ ВИЩОЇ ОСВІТИ

Гергелю Вадиму Віталійовичу

1. Тема кваліфікаційної роботи: Розробка та дослідження веб-застосунку для стеження за ментальним здоров'ям.
- Керівник роботи: д.т.н., професор Андрущак І. Є.
затверджені наказом закладу вищої освіти від «29» березня 2025 р. № 190/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи: «04» грудня 2025 р.
3. Вихідні дані до роботи технічне та програмне забезпечення ЕОМ
4. Зміст розрахунково-пояснювальної записки: аналіз проблематики прогнозування попиту та вибір методів дослідження, обґрунтування технологій і реалізацію веб-застосунку на основі регресійного аналізу, експериментальне дослідження результативності програмного забезпечення
5. Перелік графічного матеріалу 15 рисунків, 7 таблиці, 18 лістингів коду

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Андрущак І. Є.</i>		
<i>Теоретичне дослідження та практична реалізація</i>	<i>Андрущак І. Є.</i>		
<i>Експериментальне дослідження системи</i>	<i>Андрущак І. Є.</i>		
<i>Нормоконтроль</i>	<i>Повстяна Ю. С.</i>		
<i>Гарант ОП</i>	<i>Андрущак І. Є.</i>		
<i>Показник запозичень тексту</i>		___%	
<i>Академічна доброчесність</i>	<i>Андрущак І. Є.</i>		

7. Дата видачі завдання « 02 » квітня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи магістра	Строк виконання етапів роботи	Примітка
1	<i>Провести огляд літературних джерел по темі кваліфікаційної роботи</i>	<i>05.09.2025 р.</i>	
2	<i>Провести аналіз загальної проблеми і вибір напрямків дослідження</i>	<i>24.09.2025 р.</i>	
3	<i>Розробити функціональну схему роботи програмного продукту</i>	<i>01.10.2025 р.</i>	
4	<i>Описати засоби розробки об'єкта проектування</i>	<i>19.10.2025 р.</i>	
5	<i>Практична реалізація об'єкта проектування</i>	<i>26.10.2025 р.</i>	
6	<i>Розробити методичку для проведення експерименту</i>	<i>05.11.2025 р.</i>	
7	<i>Провести аналіз результатів експерименту</i>	<i>15.11.2025 р.</i>	
8	<i>Здача чистового варіанту магістерської роботи на кафедрі</i>	<i>04.12.2025 р.</i>	

Здобувач вищої освіти

_____ (підпис)

Гергель В. В.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

_____ (підпис)

Андрущак І. Є.

(прізвище, ініціали)

АНОТАЦІЯ

Гергель. В. В. Розробка та дослідження веб-застосунку для стеження за ментальним здоров'ям. Рукопис.

Кваліфікаційна робота магістра ОП «Інженерія програмного забезпечення» спеціальності 121 «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота магістра складається з вступу, 3 розділів, висновків і пропозицій, списку використаних джерел, додатків (згідно структури кваліфікаційної роботи, затвердженої кафедрою).

У першому розділі проведено аналіз стану проблематики стеження за ментальним здоров'ям, розглянуто наявні наукові підходи, існуючі програмні рішення, а також методи та засоби розробки веб-застосунків відповідного призначення. У другому розділі обгрунтовано вибір технологій, а також реалізовано програмні компоненти серверної та клієнтської частин веб-застосунку для моніторингу ментального стану користувача. У третьому розділі проведено експериментальну оцінку ефективності розробленого застосунку, описано методика проведення дослідження. У висновках узагальнено результати теоретичного й практичного дослідження.

Ключові слова: веб-застосунок, ментальне здоров'я, Nuxt ,Vue, Laravel, API.

ABSTRACT

Gergel, V. V. Development and research of a web application for monitoring mental health. Manuscript.

Master's Qualification Thesis of the Educational Program "Software Engineering" specialty 121 Software Engineering. Lutsk National Technical University. Lutsk, 2025.

The master's thesis consists of an introduction, 3 chapters, conclusions and recommendations, a list of references, and appendices (in accordance with the structure of the thesis approved by the department).

The first chapter analyzes the state of mental health monitoring, reviews existing scientific approaches, existing software solutions, as well as methods and means of developing web applications for this purpose. The second chapter justifies the choice of technologies and implements the software components of the server and client parts of the web application for monitoring the user's mental state. The third chapter provides an experimental evaluation of the effectiveness of the developed application and describes the research methodology. The conclusions summarize the results of theoretical and practical research.

Keywords: web application, mental health, Nuxt, Vue, Laravel, API.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ ЗА ТЕМОЮ РОБОТИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ.....	9
1.1 Огляд і аналіз предметної області проблеми, результатів існуючих теоретичних та експериментальних досліджень.....	9
1.2 Огляд і аналіз методів та засобів розробки веб-застосунку для стеження за ментальним здоров'ям для вирішення проблеми дослідження ...	19
1.3 Постановка завдання на кваліфікаційну роботу магістра	24
РОЗДІЛ 2 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ СТЕЖЕННЯ ЗА МЕНТАЛЬНИМ ЗДОРОВ'ЯМ	25
2.1 Обґрунтування вибору шляхів, технологій, алгоритмів і засобів вирішення поставленого завдання.....	25
2.2 Практична реалізація об'єкта проектування	27
РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ ТЕХНІЧНИХ ХАРАКТЕРИСТИК ВЕБ-ЗАСТОСУНКУ ДЛЯ СТЕЖЕННЯ ЗА МЕНТАЛЬНИМ ЗДОРОВ'ЯМ	49
3.1 Методика проведення дослідження	49
3.2 Обробка та аналіз отриманих результатів	50
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56

ВСТУП

Актуальність кваліфікаційної роботи магістра визначена тим, що у сучасних умовах постійного стресу, соціальної нестабільності та інформаційного перевантаження питання ментального здоров'я стає одним із ключових аспектів якості життя людини. Особливо гостро це проявляється в умовах війни, коли психологічний стан населення зазнає значних навантажень. Своєчасне відстеження власного емоційного стану та формування усвідомленого ставлення до психічного здоров'я сприяє покращенню адаптаційних можливостей особистості та підвищенню продуктивності в особистому й професійному житті.

Метою дослідження є розробка та дослідження веб-застосунку, який дозволяє користувачеві фіксувати, зберігати та візуалізувати інформацію про свій психологічний стан з використанням сучасних технологій веб-програмування.

Об'єкт дослідження – процес розробки веб-застосунку для моніторингу показників ментального здоров'я користувача.

Предмет дослідження – методи, засоби та технології створення веб-застосунків для збору, оброблення та представлення даних, що характеризують ментальний стан користувача.

Завданням дослідження є:

- здійснити аналітичний огляд існуючих веб-рішень для підтримки ментального здоров'я;
- обрати сучасні інструменти та технології для реалізації клієнтської та серверної частини веб-застосунку;
- розробити фронтенд-частину веб-застосунку для взаємодії користувача з системою;
- розробити структуру бази даних для збереження показників ментального стану користувача;
- реалізувати підключення та інтеграцію API між клієнтською та серверною частинами веб-застосунку;

- провести тестування працездатності веб-застосунку.

Наукова новизна отриманих результатів кваліфікаційної роботи магістра полягає у створенні структурно-функціональної моделі веб-застосунку для моніторингу ментального здоров'я, який поєднує можливість фіксації та візуалізації емоційних станів користувача, а також забезпечує можливість аналізу змін у часовому вимірі. Запропонований підхід дозволяє інтегрувати елементи самооцінки та самоспостереження у щоденну діяльність користувача.

Практичне значення кваліфікаційної роботи полягає у створенні робочого прототипу веб-застосунку, який може бути використаний для особистого контролю психологічного стану, а також у навчальних, корпоративних чи медичних установах як допоміжний інструмент моніторингу емоційного добробуту. Отримані результати можуть бути використані для подальших досліджень у галузі розробки цифрових сервісів підтримки ментального здоров'я.

Результати дослідження були сформовані в науковій статті та опубліковані в наукових журналах «Комп'ютерно-інтегровані технології: освіта, наука, виробництво». Луцьк: Луцький НТУ, 2025. Вип. № 59 та «Студентський науковий вісник». Луцьк: Луцький НТУ, 2025. Вип. № 54 [1, 2].

РОЗДІЛ 1

АНАЛІЗ ПРОБЛЕМАТИКИ ЗА ТЕМОЮ РОБОТИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Огляд і аналіз предметної області проблеми, результатів існуючих теоретичних та експериментальних досліджень

У сучасному глобалізованому світі швидкий розвиток технологій, динамічні соціальні зміни та постійна політична нестабільність створюють численні виклики для психічного здоров'я людини [3]. Психоемоційна рівноваженість сьогодні виступає не лише особистісною потребою, а й необхідною умовою активної участі у професійному, соціальному та особистому житті. Особливої гостроти ця проблема набула у зв'язку з війною в Україні, яка принесла масові втрати, розлуку, вимушене переміщення населення та посилення почуття невизначеності й тривоги. За таких умов велика кількість людей відчуває потребу у фаховій психологічній підтримці, адже травматичний досвід і щоденне емоційне напруження залишають глибокий відбиток на психічному стані особистості.

Сучасна доба характеризується надзвичайно високими темпами поширення інформації. Глобалізаційні процеси забезпечують миттєвий доступ до подій у різних куточках світу, проте водночас формують надмірне інформаційне навантаження, що негативно впливає на емоційне самопочуття. Постійний потік новин, особливо негативного змісту, здатен провокувати підвищений рівень стресу, тривожності та інші розлади психоемоційної сфери. Активне впровадження цифрових технологій, мобільних пристроїв та швидкісного інтернету докорінно змінило ритм повсякденного життя, змусивши людину постійно адаптуватися до нових умов і приймати рішення у реальному часі. Попри зручність і ефективність таких технологій, вони часто стають причиною психологічного виснаження, емоційного вигорання та загального погіршення якості життя [4].

Попит на психологічну підтримку в останні роки істотно зріс, однак її

доступність залишається обмеженою. Найбільших труднощів у цьому аспекті зазнають жителі малих населених пунктів, де бракує фахівців і матеріальних ресурсів. До того ж, значною перешкодою є соціальна стигматизація звернень до психолога, яка й досі поширена у суспільстві. Через це багато осіб уникають консультацій, що з часом лише погіршує їхній стан.

Окремої уваги заслуговує процес віртуалізації міжособистісного спілкування. Традиційні форми контактів поступово замінюються цифровими, що, з одного боку, дає змогу підтримувати зв'язок на будь-якій відстані, а з іншого – не забезпечує повноцінного емоційного обміну, необхідного для стабільного психічного стану. Соціальні мережі та месенджери дозволяють бути на зв'язку, проте не здатні компенсувати дефіцит живого спілкування та глибокої емоційної взаємодії.

Цифровізація соціального простору відкриває нові можливості для покращення доступу до психічної допомоги. Онлайн-платформи та мобільні застосунки нині виступають ефективними інструментами дистанційного надання психологічної підтримки. Вони забезпечують користувачам можливість отримати консультацію, інформаційні матеріали чи інструменти самопомоги незалежно від місця проживання. До таких сервісів належать відеоконференції, чат-боти, онлайн-консультації тощо, що особливо корисно для населення віддалених регіонів.

Веб-застосунки пропонують широкий спектр функцій для самостійного контролю психічного стану – від релаксаційних практик і медитацій до ведення щоденників настрою чи тренування навичок управління стресом. Крім цього, цифрові платформи активно використовуються для популяризації культури психічного благополуччя: проведення освітніх кампаній, семінарів, тренінгів і вебінарів сприяє підвищенню рівня поінформованості населення та зниженню стигматизації звернень за допомогою.

Особливо цікавим є аналіз уже існуючих веб-платформ для підтримки ментального здоров'я, які функціонують на українському ринку.

Першим прикладом є веб-платформа «HedePy» – це сучасний сервіс для онлайн-консультацій із сертифікованими психологами. Основна сторінка виконана у спокійній кольоровій гамі, що сприяє створенню відчуття довіри та психологічного комфорту. Інтерфейс простий та зручний, можна пройти коротке опитування, обрати напрям допомоги й забронювати сеанс. Після авторизації відкриваються розширені можливості – перегляд профілів психологів, запис на консультацію, доступ до тематичних матеріалів і тренінгів. Важливо зазначити, що всі консультації на платформі є платними, що зумовлює високий рівень відповідальності з боку фахівців та якість наданих послуг. Приклад головної сторінки зображено на рисунку 1.1.

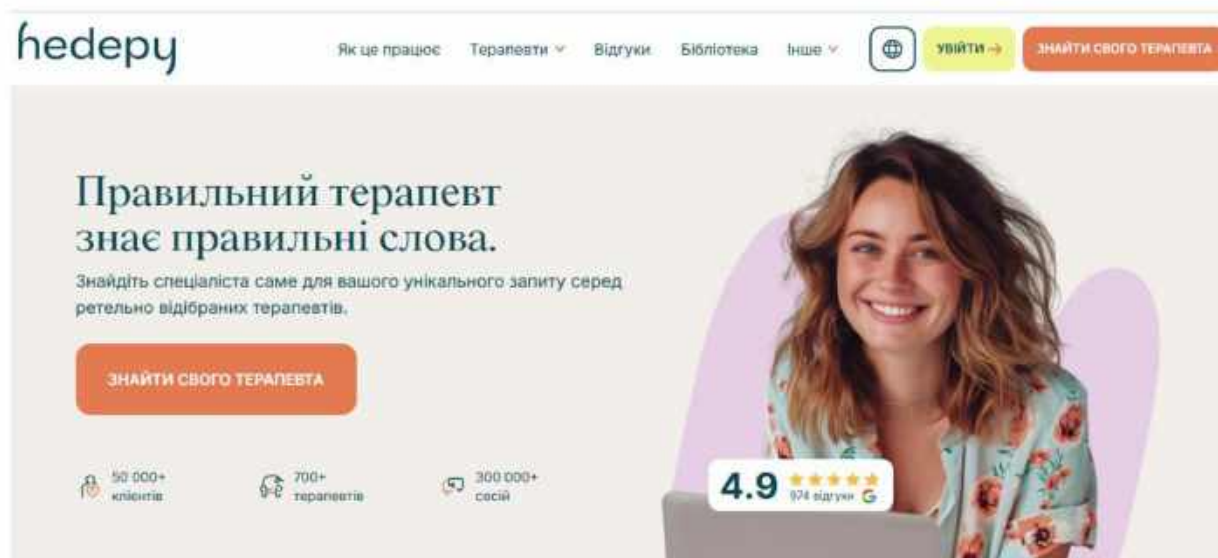


Рисунок 1.1 – Головна сторінка «HedePy» [5]

Іншим прикладом є платформа «HowAreU», яка пропонує користувачам пройти психологічний тест для первинної оцінки емоційного стану. Сайт вирізняється простим, мінімалістичним дизайном, що дозволяє швидко зорієнтуватися в основному функціоналі. Після проходження тесту користувач отримує можливість звернутися до фахівця та записатися на консультацію. Платформа працює на платній основі, надаючи доступ до онлайн-сеансів, навчальних матеріалів і порад з управління стресом. Приклад головної сторінки зображено на рисунку 1.2.

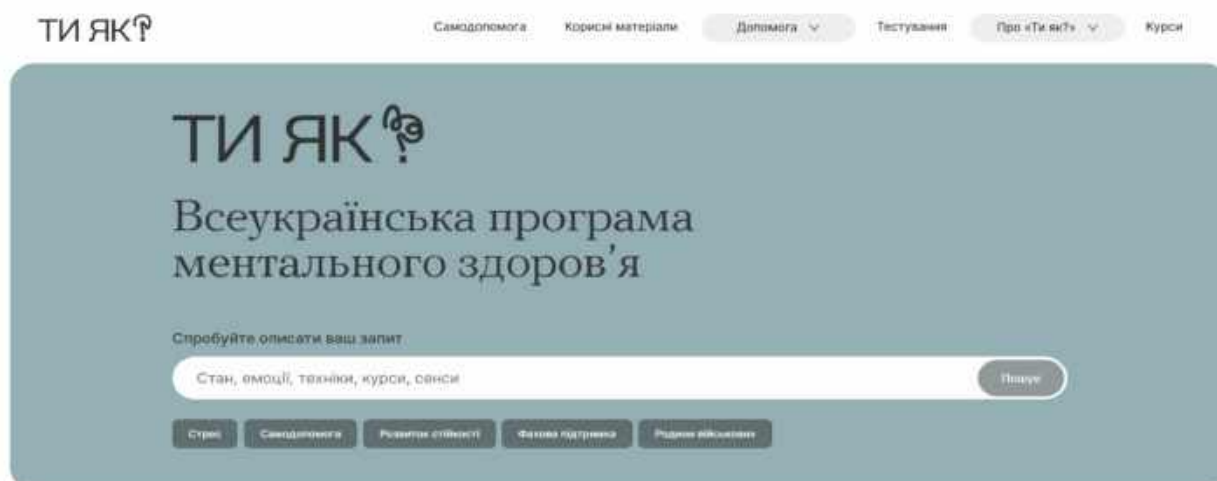


Рисунок 1.2 – Головна сторінка «HowAreU» [6]

Ще одним прикладом є «MentalCode» – комплексна платформа, що поєднує індивідуальні консультації психологів, навчальні програми з ментального добробуту та корпоративні рішення для бізнесу. Дизайн сайту виконано у стриманих тонах, що відповідає професійному стилю. Особливістю є авторизація користувача, після якої відкриваються розширені можливості бронювання сеансів, доступ до тренінгів, тестів і матеріалів із розвитку емоційної стійкості. Усі послуги надаються на платній основі, що забезпечує персоналізований підхід і високий рівень якості обслуговування. Приклад головної сторінки зображено на рисунку 1.3.

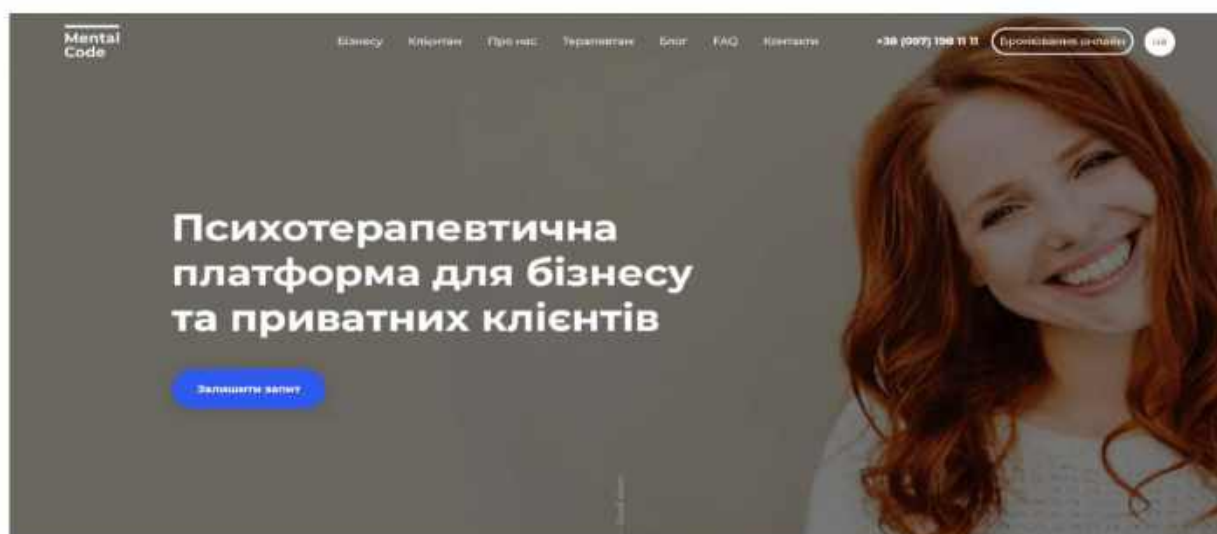


Рисунок 1.3 – Головна сторінка «MentalCode» [7]

Порівняльний аналіз показує, що всі три веб-платформи орієнтовані на надання платних послуг із психологічної підтримки, проте мають різні акценти у своїй діяльності. «HedePy» робить акцент на індивідуальних консультаціях із кваліфікованими психологами; «HowAreU» пропонує первинну самодіагностику через тестування та подальший запис на сеанс; «MentalCode» поєднує індивідуальну і корпоративну підтримку, сприяючи розвитку культури турботи про ментальне здоров'я серед працівників організацій.

Розвиток цифрових рішень сприяє також оптимізації традиційної системи охорони здоров'я. Використання попередніх онлайн-консультацій дає змогу швидше визначити потреби пацієнта й спрямувати його до відповідного спеціаліста, тим самим підвищуючи ефективність розподілу ресурсів і скорочуючи час очікування. Забезпечення конфіденційності та безпеки персональних даних користувачів стає одним із головних пріоритетів розробників таких застосунків, що сприяє зростанню довіри до цифрових форм психологічної допомоги.

Перспективи подальшого розвитку технологій у сфері психічного здоров'я є надзвичайно широкими. Інтеграція мобільних сервісів у загальну систему охорони здоров'я може створити єдину мережу підтримки, у якій цифрові рішення відіграватимуть роль першої лінії контакту з людиною, що потребує допомоги. Застосування штучного інтелекту відкриває можливості для аналізу емоційного стану користувачів і надання персоналізованих рекомендацій [8]. Крім того, функціонал соціальних платформ усередині мобільних застосунків створює простір для взаємопідтримки між людьми з подібними труднощами, що доповнює професійну психологічну роботу.

Такі застосунки допомагають користувачам відстежувати власний емоційний стан, фіксувати зміни настрою та аналізувати фактори, що впливають на їхнє самопочуття. Зокрема, користувачі отримують доступ до коротких психоосвітніх модулів, які навчають розпізнавати ознаки емоційного виснаження, страху чи паніки та застосовувати ефективні техніки подолання

стресу. Це особливо важливо для людей, які проживають у зонах бойових дій або зазнали травматичного досвіду втрати, розлуки чи руйнування дому.

Однією з ключових переваг цифрових платформ є можливість анонімного користування. Багато українців і досі уникають звернення до психологів через суспільну стигматизацію, тому онлайн-сервіси створюють безпечний простір для самопомоги без страху осуду. Анонімність, простота використання та конфіденційність даних сприяють тому, що навіть користувачі з високим рівнем тривожності чи недовіри до фахівців готові спробувати цифрові методи відновлення психічного стану.

Аналіз основних стресових факторів, що впливають на українців у воєнних умовах, свідчить, що найсильніше на психіку діють відчуття небезпеки, втрата близьких, постійна невизначеність, інформаційний тиск і соціальна ізоляція. Тривале перебування у стані напруження призводить до зниження здатності до концентрації, розладів сну, емоційного вигорання та посттравматичних реакцій. У цьому контексті цифрові засоби стають інструментом, який дозволяє людині поступово відновлювати контроль над своїм психоемоційним станом.

Важливо, що розробники таких платформ враховують культурні, мовні та соціальні особливості користувачів. Україномовний контент, адаптований до реалій війни, допомагає створити відчуття розуміння та підтримки, що значно підвищує ефективність психологічної допомоги. Цифрові програми не лише сприяють зниженню тривожності, а й формують навички саморефлексії, що допомагає людині зміцнити внутрішні ресурси й розвинути стійкість до стресу [9].

Попри очевидні переваги цифрових рішень для підтримки психічного здоров'я, однією з найважливіших проблем залишається забезпечення конфіденційності персональних даних користувачів. У сучасних умовах більшість онлайн-платформ і мобільних застосунків акумулюють значний обсяг чутливої інформації – від результатів психологічних тестів до індивідуальних записів про емоційний стан, поведінку чи навіть фізичну активність користувача. Неналежне поводження з такими даними може створювати ризики

несанкціонованого доступу або витоку інформації, що підриває довіру до цифрових форм психічної допомоги.

Дослідження, проведене Iwaya у 2023 році, показало, що значна частина мобільних застосунків, спрямованих на покращення психічного благополуччя, не відповідає базовим вимогам безпеки й прозорості щодо обробки персональних даних. Автори виявили, що багато застосунків збирають інформацію, яка виходить за межі необхідного для надання послуг, наприклад, доступ до геолокації, контактів або соціальної активності користувача. Крім того, політики конфіденційності часто подані в складній юридичній формі, що унеможлиблює їхнє повне розуміння пересічним користувачем.

Проблема ускладнюється тим, що значна частина таких сервісів розробляється приватними компаніями, а не медичними чи академічними установами, тому контроль за дотриманням етичних стандартів може бути обмеженим. У сфері психічного здоров'я, де довіра є критично важливою умовою ефективної взаємодії, будь-яке порушення конфіденційності здатне спричинити серйозні психологічні наслідки для користувачів. З огляду на це, питання безпеки набуває особливого значення в умовах війни, коли багато українців змушені користуватися онлайн-ресурсами як єдиним способом отримання психологічної підтримки.

Ще одним важливим аспектом є ризик нецільового використання даних, які збираються під час роботи таких застосунків. Відомі випадки, коли інформація з платформ для психічного здоров'я передавалася третім сторонам, зокрема рекламним компаніям, що викликає етичні сумніви щодо їхньої діяльності. Тому сучасні розробники повинні дотримуватися принципів «privacy by design» – тобто впроваджувати механізми захисту даних на всіх етапах створення та функціонування цифрового продукту.

Особливої актуальності ці питання набувають у контексті використання технологій штучного інтелекту для аналізу емоційного стану користувачів. Алгоритми, які збирають голосові, текстові чи поведінкові сигнали, можуть надавати цінну інформацію для оцінки психічного стану, але водночас

створюють додаткові ризики вторгнення у приватне життя. Відсутність чітких етичних і правових механізмів регулювання таких систем може призвести до зловживань або помилкової інтерпретації результатів.

В умовах воєнного та поствоєнного суспільства, де психологічна вразливість населення підвищена, питання етичного використання даних набуває стратегічного значення. Саме тому важливо, щоб цифрові рішення у сфері психічного здоров'я поєднували наукову достовірність, технологічну інноваційність і найвищі стандарти захисту персональних даних користувачів [10].

Дослідження Лінардона у 2024 році показало, що використання мобільних застосунків для підтримки психічного здоров'я здатне помітно зменшувати прояви депресії та генералізованої тривожності. Зокрема, результати свідчать про середнє зниження показників депресії на 6,6 балів за шкалою «PHQ-9» та рівня тривожності на 4,4 бали за шкалою «GAD-7» після регулярного користування додатками. Це підтверджує, що цифрові інструменти можуть ефективно підтримувати психоемоційний стан користувачів, особливо у випадках обмеженого доступу до очних консультацій психологів [10].

Як ще один приклад можна навести програму «Sleep2Ukraine», розроблену для допомоги людям, які постраждали від війни. Ця платформа, заснована на принципах когнітивно-поведінкової терапії для лікування безсоння, показала високу ефективність у покращенні якості сну та зменшенні проявів депресії й тривожності серед її користувачів. Люди, які регулярно застосовували програму, відзначали суттєве покращення свого емоційного стану та зниження рівня стресу [10].

Особливої уваги потребує інтеграція систем штучного інтелекту, які здатні аналізувати емоційний стан користувачів та надавати персоналізовані рекомендації. Для забезпечення етичності використання таких технологій необхідно розробляти чіткі алгоритми обробки даних, які мінімізують ризики помилкової інтерпретації або зловживань. Наприклад, алгоритми можуть

працювати виключно з анонімізованими даними або обмежувати доступ до чутливої інформації лише авторизованим фахівця [11].

Водночас використання таких цифрових засобів пов'язане з певними обмеженнями. Технічні проблеми, як-от низька швидкість інтернету або обмежена доступність смартфонів, психологічні фактори, включаючи низьку мотивацію чи складність адаптації до технологій серед старшого покоління, а також соціальні аспекти, як недовіра до платформ через ризики конфіденційності, можуть знижувати ефективність цих методів.

Нижче представлено узагальнений порівняльний аналіз основних стрес-факторів, цифрових інструментів для їхньої компенсації та обмежень зображений у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз основних стрес-факторів

Стрес-фактор	Цифрові інструменти підтримки	Можливі обмеження використання
Постійна тривога та стрес	Медитації, релаксаційні вправи, щоденники настрою	Низька мотивація, складність регулярного використання
Втрата близьких, соціальна ізоляція	Онлайн-групи підтримки, відео-консультації	Недовіра до платформ, емоційна закритість користувачів
Інформаційне перевантаження	Фільтри контенту, цифрові паузи, психологічні модулі	Обмежені можливості персоналізації, недостатня обізнаність про інструменти
Порушення сну	Програми для релаксації перед сном	Низька доступність гаджетів або інтернету
Емоційне вигорання	Комплексні онлайн-програми з когнітивно-поведінковою терапією	Складність адаптації до технологій, психологічний опір змінам

Як видно з наведеної таблиці, різні цифрові інструменти мають специфічний вплив на основні стресові фактори, з якими стикаються українці в умовах війни. Щоденники настрою та трекери емоцій дозволяють відстежувати зміни психоемоційного стану, медитації та релаксаційні вправи знижують рівень тривоги і напруження, а онлайн-групи підтримки та форуми сприяють соціальній

взаємопідтримці та зменшенню відчуття ізоляції. Разом із тим, ефективність використання цих інструментів значною мірою залежить від технічних можливостей, мотивації користувачів і довіри до цифрових платформ.

Сучасні цифрові платформи для підтримки психічного здоров'я відкривають можливості не лише для індивідуальної самопомоги, а й для створення спільнот взаємопідтримки. Через онлайн-форуми, групи у месенджерах і соціальних мережах користувачі можуть обмінюватися досвідом, ділитися стратегіями подолання стресу та підтримувати одне одного у кризових ситуаціях. Такий соціальний компонент не лише зменшує почуття ізоляції, але й підсилює мотивацію до регулярного використання цифрових ресурсів [10].

Варто відзначити, що цифрові інструменти здатні адаптуватися до індивідуальних потреб користувача. Мобільні застосунки часто пропонують персоналізовані рекомендації на основі анкетування, щоденників настрою чи поведінкових патернів. Наприклад, користувачі можуть отримувати сповіщення про необхідність виконати дихальні вправи, медитацію чи коротку фізичну активність залежно від рівня тривожності, зафіксованого додатком. Це дозволяє створити «інтелектуальну підтримку», яка реагує на стан користувача у реальному часі [10].

Разом з тим, ефективність цифрових платформ часто залежить від мотивації та дисципліни користувача. Деякі люди можуть швидко втрачати інтерес або відчувати розчарування через складність інтерфейсу чи відсутність швидких результатів. Для старшого покоління або людей із низькою цифровою грамотністю це стає серйозною перешкодою. Тому розробники все частіше приділяють увагу інтуїтивності дизайну, наочності інструкцій та інтеграції елементів ігрової мотивації, що стимулює регулярне використання застосунку.

Цифрові рішення також дозволяють поєднувати різні підходи у психотерапії. Наприклад, когнітивно-поведінкові техніки можна поєднувати з медитаціями, веденням щоденника та аудіо або відео вправами. Такий мультимедійний підхід дозволяє комплексно впливати на психоемоційний стан, формуючи у користувача навички саморегуляції та емоційної стійкості.

Багато програм пропонують короткі та доступні сесії, що дає змогу інтегрувати практики у повсякденне життя навіть у стресових умовах.

Важливою складовою є збір та аналіз даних для покращення сервісу. Анонімізовані дані про активність користувачів дозволяють розробникам визначати, які функції застосунку є найбільш ефективними, а які потребують доопрацювання. Це створює можливість для постійного вдосконалення цифрових інструментів та підвищення їхньої практичної цінності для користувачів, одночасно зберігаючи конфіденційність і безпеку персональної інформації.

Таким чином, сучасні цифрові платформи для підтримки психічного здоров'я демонструють значний потенціал у подоланні наслідків стресових ситуацій, зокрема тих, що виникають унаслідок воєнних подій. Вони поєднують доступність, адаптованість до культурного та мовного контексту користувачів, інтерактивність та можливість персоналізованої підтримки. Водночас ефективність таких інструментів безпосередньо залежить від дотримання принципів безпеки, прозорості та етичності обробки даних. Інтеграція технологій шифрування, багаторівневої аутентифікації, а також навчання користувачів основам цифрової гігієни стає ключовою умовою формування довіри та забезпечення сталого використання цих сервісів у довгостроковій перспективі.

1.2 Огляд і аналіз методів та засобів розробки веб-застосунку для стеження за ментальним здоров'ям для вирішення проблеми дослідження

У сучасній веб-розробці існує широкий спектр технологій, інструментів та фреймворків, які відкривають можливість створювати надійні, швидкі та функціонально насичені онлайн-сервіси. Вибір правильного стеку технологій є критично важливим для успіху проєкту, особливо коли йдеться про застосунок, що працює з конфіденційними даними та потребує складної візуалізації, як-от система для стеження за ментальним здоров'ям. Для вирішення проблеми

дослідження розробки веб-застосунку для стеження за ментальним здоров'ям було проаналізовано ключові технології фронтенду, бекенду, стилізації та управління контентом.

Vue.js – це прогресивний JavaScript-фреймворк, призначений для створення інтерактивних користувацьких інтерфейсів і односторінкових застосунків. Його головна перевага полягає в інкрементальній адаптивності, яка дозволяє легко інтегрувати Vue у будь-який проєкт. Основними плюсами цього фреймворку є простота освоєння та швидкість навчання, що забезпечується інтуїтивним, HTML-орієнтованим синтаксисом у форматі однофайлових компонентів. Це сприяє зрозумілості структури коду та пришвидшує процес розробки. Окрім цього, Vue.js має високу продуктивність завдяки використанню віртуального «DOM», який мінімізує кількість безпосередніх маніпуляцій із реальним «DOM», що позитивно впливає на швидкодію застосунку. Також Vue.js характеризується гнучкістю архітектури, підтримуючи як традиційний «Options API», так і сучасний «Composition API», що дозволяє обирати оптимальний підхід до структурування коду залежно від складності проєкту [12].

Nuxt 3 – це високопродуктивний, відкритий фреймворк, побудований на основі Vue.js, який значно спрощує створення застосунків. Його головна цінність полягає у наданні готової архітектурної оболонки з налаштованими параметрами, що дає змогу розробникам зосередитися на реалізації бізнес-логіки. Однією з ключових особливостей Nuxt є автоматична підтримка «Server-Side Rendering» – технології, за якої початковий HTML-код генерується на сервері й передається клієнту. Це суттєво покращує SEO, адже пошукові системи бачать повністю згенерований контент, а також зменшує час відображення сторінки, що підвищує якість користувацького досвіду [13].

Ключові особливості Nuxt 3:

- автоматичний роутинг і модульна система. Маршрути створюються автоматично на основі структури папки «pages», що прискорює розробку, а модульна система дозволяє швидко підключати додаткові функції;

- оновлена архітектура та підвищена продуктивність. У версії Nuxt 3 покращено досвід розробників, оптимізовано «Hot Module Replacement» і введено каталог «app/», що забезпечує кращу структурування проєкту;

- інтелектуальна система отримання даних. Автоматично кешує запити з однаковими ключами, зменшує навантаження на сервер і запобігає дублюванню запитів та витокам пам'яті.

React – це популярна JavaScript-бібліотека для побудови користувацьких інтерфейсів, яка швидко стала стандартом у веб-розробці. Її концепція базується на компонентному підході та застосуванні «Virtual DOM», що забезпечує високу швидкість і ефективність оновлення UI. React використовує «JSX» – синтаксис, який поєднує JavaScript і HTML [14].

Next.js – це потужний фреймворк, створений на основі React, який розширює його можливості до повноцінної фулстек-розробки. Він є провідним інструментом для створення масштабованих, SEO-оптимізованих і високопродуктивних застосунків [15].

Основна перевага Next.js полягає у вбудованій підтримці різних стратегій рендерингу: «Server-Side Rendering» і «Static Site Generation», які забезпечують швидке завантаження та ефективну індексацію сторінок. Next.js також має файлову маршрутизацію, що автоматично генерує маршрути, і систему API Routes, завдяки чому розробники отримують єдине середовище для створення як фронтенду, так і бекенду, що робить Next.js потужним рішенням для сучасних веб-платформ.

Angular – це повноцінний, відкритий, MVC-фреймворк, розроблений та підтримуваний компанією Google, який спеціалізується на побудові великомасштабних, корпоративних клієнтських застосунків. На відміну від React чи Vue.js, Angular є закінченим рішенням, яке постачається з власним потужним інструментарієм, включаючи маршрутизатор, HTTP-клієнт та механізми управління формами.

Ключовою архітектурною перевагою є його орієнтація на мову TypeScript, яка забезпечує статичну типізацію та дозволяє виявляти помилки ще на етапі

компіляції, що суттєво підвищує надійність і підтримуваність коду у великих командах. Angular використовує патерн «Залежність від Впровадження», що сприяє створенню модульної, тестованої та легко керованої архітектури. Хоча він є частиною популярних стеків, Angular часто використовується як самостійний, всеосяжний інструмент, що має вищий поріг входу, але забезпечує неперевершену узгодженість і передбачуваність коду [16].

Less – це динамічна мова стилів, або, простіше кажучи, CSS-препроцесор, схожий на SCSS/Sass. Його основне завдання – розширити стандартний CSS, додаючи такі функції, як змінні, міксини та вкладені правила. Це дозволяє розробникам писати чистіший, багаторазово використовуваний та модульний код стилів, який легше обслуговувати у великих проєктах. Перед застосуванням у браузері Less-код має бути скомпільований у чистий CSS [17].

Tailwind CSS – це орієнтований на утиліти CSS-фреймворк. Він працює за принципово іншим підходом, ніж препроцесори. Замість того, щоб писати традиційні CSS-блоки та давати їм семантичні імена, розробники застосовують готові утилітарні класи безпосередньо в HTML-розмітці. Такий підхід значно прискорює створення інтерфейсу, мінімізує необхідність перемикання між HTML та CSS-файлами та дозволяє швидко створювати унікальні дизайни без написання власного CSS [18].

SCSS – це найбільш популярний CSS-препроцесор, який розширює можливості стандартного CSS, додаючи функції, притаманні мовам програмування. Він використовує синтаксис, максимально наближений до звичайного CSS, але доповнений ключовими особливостями: змінними для керування кольорами та розмірами, міксинами для створення багаторазових блоків стилів, функціями та успадкуванням [19]. Використання SCSS значно підвищує ефективність написання стилів, їх читабельність та обслуговування у великих і складних кодових базах, що є критично важливим для проєктів зі складними вимогами до візуалізації даних.

Laravel – це один з найпопулярніших, потужних PHP-фреймворків, що

використовує архітектурний патерн «Model-View-Controller» [20]. Він розроблений для швидкої та ефективної побудови надійних веб-додатків.

Ключові особливості Laravel:

- архітектура «MVC». Чітке розділення логіки, даних та представлення, що підвищує читабельність, масштабованість та ремонтпридатність коду;
- надає простий, об'єктно-орієнтований спосіб взаємодії з базою даних, який спрощує виконання складних запитів;
- вбудовані засоби безпеки. Laravel пропонує готові рішення для захисту від поширених вразливостей, а також систему автентифікації та авторизації;
- спрощена обробка черг. Дозволяє відкладати трудомісткі завдання для виконання у фоновому режимі, що підвищує продуктивність застосунку.

Django – це сучасний і зручний веб-фреймворк на Python, який працює за принципами «все в комплекті» та DRY (Don't Repeat Yourself). Він одразу містить багато готових інструментів, тож розробнику не потрібно додавати безліч сторонніх бібліотек або дублювати код [21].

Ключові особливості Django:

- вбудований «ORM» (Object-Relational Mapping). Забезпечує інтуїтивну, об'єктно-орієнтовану взаємодію з базою даних, спрощуючи виконання складних запитів та прискорюючи розробку серверної логіки;
- автоматично згенерована адміністративна панель. Надає готовий, повнофункціональний інтерфейс для керування даними без необхідності додаткової розробки;
- комплексні засоби безпеки. Включають захист від поширених вразливостей, механізми автентифікації та авторизації, що є критично важливим для застосунків, які працюють із конфіденційною інформацією;
- підтримка Django REST Framework. Дозволяє швидко створювати надійні та безпечні REST API, що робить Django ефективним вибором для сучасних сервісів і мобільних бекендів [22];

– масштабованість та модульність. Структура проєкту сприяє легкому розширенню функціоналу, а велика екосистема пакетів та активна спільнота забезпечують тривалу підтримку та наявність перевірених рішень.

1.3 Постановка завдання на кваліфікаційну роботу магістра

У сучасних умовах зростання психологічного виснаження, емоційної напруги та поширення ментальних розладів особливо актуальними стають цифрові інструменти, здатні допомогти людині аналізувати свій емоційний стан та підтримувати психологічний баланс. Веб-застосунки для стеження за ментальним здоров'ям надають можливість користувачам фіксувати зміни настрою, працювати зі щоденниками емоцій і формувати загальну картину психоемоційного благополуччя.

Враховуючи цілі, призначення та функціональні особливості майбутнього веб-застосунку, було сформовано перелік завдань, що мають бути виконані в межах кваліфікаційної роботи:

- здійснити аналітичний огляд існуючих веб-рішень для підтримки ментального здоров'я;
- обрати сучасні інструменти та технології для реалізації клієнтської та серверної частини веб-застосунку;
- розробити фронтенд-частину веб-застосунку для взаємодії користувача з системою;
- розробити структуру бази даних для збереження показників ментального стану користувача;
- реалізувати підключення та інтеграцію API між клієнтською та серверною частинами веб-застосунку;
- провести тестування працездатності веб-застосунку.

РОЗДІЛ 2

ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ СТЕЖЕННЯ ЗА МЕНТАЛЬНИМ ЗДОРОВ'ЯМ

2.1 Обґрунтування вибору шляхів, технологій, алгоритмів і засобів вирішення поставленого завдання

Для розробки веб-застосунку, який вимагає високої продуктивності, надійності та чудового користувацького досвіду, обрано технологічний стек, що складається з Nuxt 3 (на базі Vue.js) для фронтенду, SCSS для стилізації та Laravel (на базі PHP) для бекенду. Такий вибір забезпечує ефективну розробку, масштабованість та міцну архітектуру. Обраний стек технологій є збалансованим, високопродуктивним та безпечним поєднанням інструментів, що повністю відповідає вимогам створення веб-платформи для стеження за ментальним здоров'ям. Обґрунтування вибору спирається на три ключові напрямки, які забезпечують успішну реалізацію проєкту. Додатково враховано специфіку предметної області, яка передбачає роботу з конфіденційною інформацією, потребу у високій швидкодії інтерфейсу, а також наявність складних механізмів аналітики та візуалізації. Саме тому обрані технології мають не лише відповідати сучасним стандартам, а й забезпечувати стабільність, безпеку та можливість подальшого масштабування системи відповідно до зростаючих потреб користувачів.

Nuxt 3 забезпечує високу продуктивність і чудовий користувацький досвід завдяки автоматичному серверному «рендерингу», що мінімізує час до відображення контенту та гарантує відмінний SEO. Фреймворк надає автоматичний «роутинг», структуровану архітектуру та оптимізовані механізми отримання даних, що значно прискорює розробку. У порівнянні з іншими фреймворками, такими як React/Next.js чи Angular, Nuxt характеризується нижчою кривою навчання, високою швидкістю розробки та орієнтованістю на швидкі, SEO-оптимізовані застосунки. Це добре ілюструє порівняння, продемонстроване у таблиці 2.1.

Таблиця 2.1 – Порівняння ключових JavaScript-фреймворків

Характеристика	Vue.js / Nuxt	React / Next.js	Angular
Крива Навчання	Низька	Середня	Висока.
Метод рендерингу	SSR, SSG, SPA завдяки Nuxt	SSR, SSG, SPA завдяки Next.js	SPA
Розмір екосистеми	Велика спільнота	Найбільша спільнота	Велика спільнота
Швидкість розробки	Висока	Висока	Середня
Підходить для	Швидко розвинуті, високопродуктивні, SEO-орієнтовані застосунки	Великі проекти з потребою в максимальній гнучкості	Корпоративні, великомасштабні, надійні SPA

Зважаючи на складність інтерфейсу й потребу у візуалізації даних про ментальне здоров'я, важливою є чиста та модульна система стилів. SCSS дозволяє створювати семантичні, добре структуровані та масштабовані стилі завдяки змінним, міксінам та логічній організації коду. На відміну від Tailwind CSS, де HTML часто перенасичений утилітарними класами, SCSS забезпечує кращу читабельність та гнучкість. Порівняно з Less, SCSS є популярнішим і має ширшу підтримку. Це демонструє відповідне порівняння, продемонстроване у таблиці 2.2.

Таблиця 2.2 – Порівняння підходів до стилізації

Характеристика	SCSS	Tailwind	Less
Мета	Організація та перевикористання логіки стилів	Швидке прототипування та дизайн	Організація та перевикористання логіки стилів
Семантичність CSS	Висока. Стилі чітко відокремлені від HTML	Низька. Стилі інлайнові, HTML перевантажений класами	Висока, аналогічно SCSS
Масштабованість	Висока	Середня	Висока
Швидкість розробки	Висока завдяки змінним, міксінам	Дуже висока завдяки готовим утиліті-класам	Середня

Laravel забезпечує надійну роботу з конфіденційними даними завдяки вбудованим механізмам безпеки, ефективному «ORM Eloquent» та чіткій структурі «MVC». Важливою перевагою є система черг «Artisan Queue», яка дозволяє виконувати асинхронні процеси у фоновому режимі без навантаження на основний додаток. У порівнянні з Django, який потребує сторонніх рішень для асинхронної обробки, Laravel пропонує більш просту та гнучку інтеграцію для API-орієнтованих систем. Це можна побачити в порівнянні, продемонстроване у таблиці 2.3.

Таблиця 2.3 – Порівняння ключових бекенд-фреймворків

Характеристика	Laravel	Django
Система Черг	Вбудована	Потребує зовнішніх інструментів
API	Чистий RESTful-підхід	Django REST Framework
Адмін-панель	Потребує додаткових пакетів або ручної розробки	Автоматично генерується, повнофункціональна
Утиліти фреймворку	Дуже потужні для автоматизації рутини	Стандартний набір команд
Фокус застосування	Швидке створення надійних API та веб-застосунків, що вимагають фонові обробки	Контентно-орієнтовані платформи, де потрібна готова адмін-панель

Таким чином, комбінація Nuxt 3, SCSS і Laravel дозволяє створити збалансовану архітектуру, яка поєднує продуктивний фронтенд, модульні стилі та надійний бекенд. Такий стек технологій забезпечує високу швидкодію, безпеку, масштабованість і довгострокову підтримуваність, що робить його оптимальним для розробки сучасного веб-застосунку з функціями моніторингу ментального здоров'я.

2.2 Практична реалізація об'єкта проектування

Для успішної та ефективної розробки запланованого веб-застосунку для стеження за ментальним здоров'ям, було обрано сучасний та збалансований стек

технологій. Кожен із програмних продуктів: Nuxt 3, SCSS та Laravel – відіграє ключову роль у створенні високоякісного, швидкого та підтримуваного застосунку.

Розробка веб-додатку передбачає виконання низки ключових етапів, кожен із яких формує фундамент для побудови цілісної та стабільної системи. У межах даного проєкту основну увагу було зосереджено на трьох важливих складових: створенні фронтенд частини застосунку, розробці його бекенд логіки та подальшому підключенні API для забезпечення повноцінної взаємодії між усіма елементами системи.

Для демонстрації логіки переходів між сторінками створено навігаційну схему, подану на рисунку 2.1. Вона показує, які розділи сервісу стають доступними залежно від того, чи авторизований користувач, чи має він акаунт, а також які сторінки відкриваються в кожному з можливих сценаріїв взаємодії.

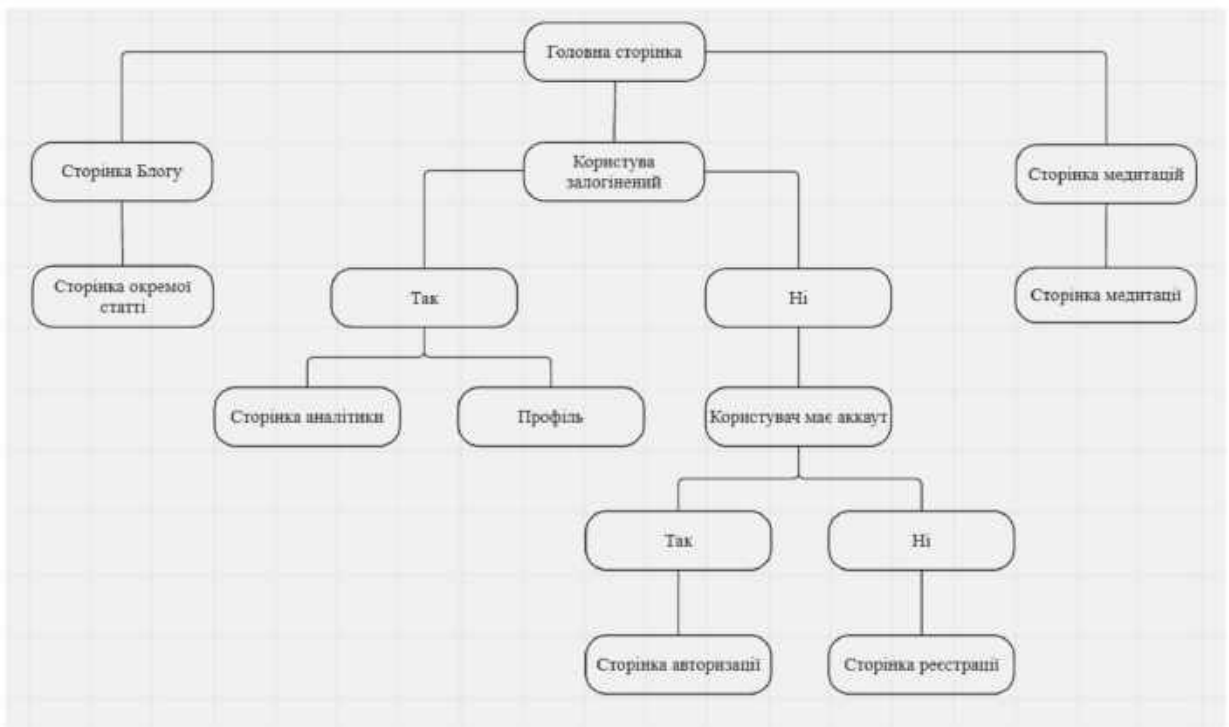


Рисунок 2.1 – Схема веб-застосунку

Першим етапом розробки було створення фронтенд частини застосунку. Для цього першим кроком ініціалізовано проєкт за допомогою команди, що

продемонстрована в лістингу 2.1, яка автоматично формує структуру директорій. Це дозволяє уникнути ручної конфігурації та забезпечує оптимальні налаштування для продуктивної роботи.

Лістинг 2.1 – команда для ініціалізації Nuxt проекту

```
npm create nuxt@latest <project-name> -- -t v3
```

Кінець лістингу 2.1

В результаті отримуємо створений проект з базовою архітектурою, яка продемонстрована на рисунку 2.2, що забезпечує впорядковану структуру файлів та дозволяє швидко розпочати подальшу розробку функціональних модулів.

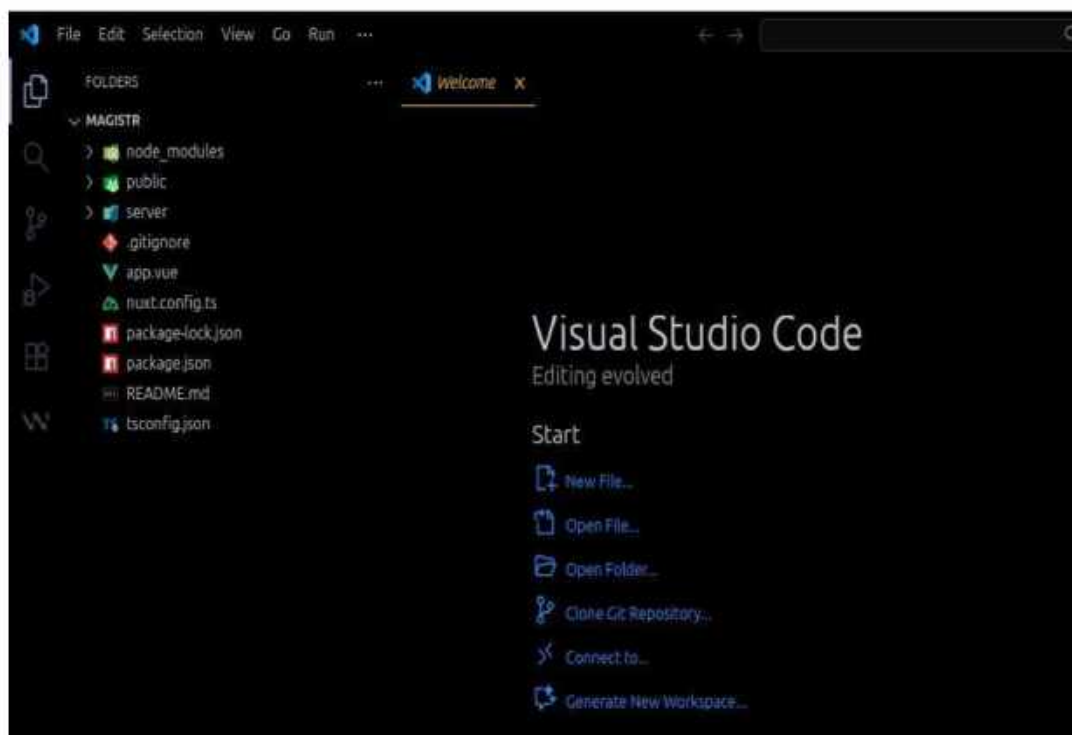


Рисунок 2.2 – Базова архітектура Nuxt

Щоб процес верстки був максимально зрозумілим і структурованим, була розроблена модульна архітектура проекту. Такий підхід забезпечує логічне групування елементів, спрощує командну роботу і мінімізує дублювання стилів та функціоналу, що продемонстровано на рисунку 2.3.

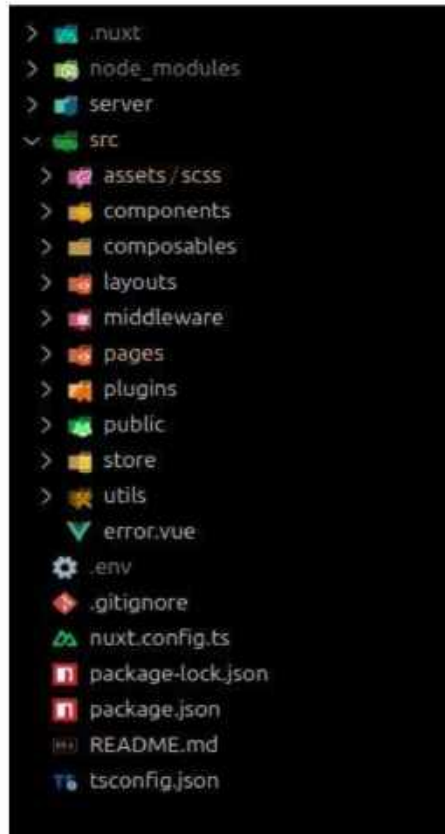


Рисунок 2.3 – Загальна архітектура проєкту

Додатковою перевагою централізованого підходу є підвищення однорідності інтерфейсу: усі компоненти використовують спільні визначення кольорів, шрифтів та відступів, що мінімізує ризик появи неузгоджених стилів. Це також забезпечує легкість у подальшому розширенні проєкту – розробнику достатньо змінити значення змінної у центральному файлі, щоб оновлення відобразилося в усіх пов'язаних елементах.

Для підвищення гнучкості стилізації та зменшення кількості повторюваного коду були впроваджені глобальні змінні для шрифтів, кольорів та базових стилів. Вони зберігаються у окремих файлах стилів та підключаються у глобальних стилях Nuxt 3. Використання змінних дозволяє оперативно змінювати візуальну складову проєкту без необхідності переписувати стилі у кожному компоненті окремо, що значно спрощує підтримку та розвиток системи, що подано на рисунку 2.4.

```

:root {
  //font
  --font-mohave-reg-16: normal 400 16rem/130% "Mohave", sans-serif;
  --font-mohave-med-16: normal 500 16rem/130% "Mohave", sans-serif;
  --font-mohave-med-28: normal 500 20rem/130% "Mohave", sans-serif;
  --font-mohave-med-25: normal 500 25rem/130% "Mohave", sans-serif;
  --font-mohave-med-30: normal 500 30rem/130% "Mohave", sans-serif;
  --font-mohave-sem-35: normal 600 35rem/130% "Mohave", sans-serif;
  --font-mohave-sem-30: normal 600 30rem/130% "Mohave", sans-serif;
  --font-mohave-bold-30: normal 700 30rem/130% "Mohave", sans-serif;
  --font-mohave-bold-28: normal 700 28rem/130% "Mohave", sans-serif;
  --font-mohave-bold-50: normal 700 50rem/130% "Mohave", sans-serif;
  --font-mohave-bold-150: normal 700 150rem/130% "Mohave", sans-serif;

  //color
  --grey: #ced2da;
  --bg: #ecdfd2;
  --bgDark: #1c1b1a;
  --black: #111111;
  --black-1: #4a4741;
  --black-2: rgba(59, 34, 34, 0.7);
  --white: #ffffff;
  --white-1: #f7f1e8;
  --white-2: #f5e6e1;
  --white-3: #d9caba;
  --blue: #53a1ee;
  --blue-1: #8cb9bc;
  --blue-2: #4e7073;
  --blue-3: #6280c2;
  --red: #ff0000;
  --green-1: #c1cfa1;
  --green-2: #284e32;
}

```

Рисунок 2.4 – Структура змінних стилів у проєкті

Для покращення взаємодії між сторінками та елементами конструктора були впроваджені плавні анімації переходів. У глобальних налаштуваннях Nuxt додано параметр, який продемонстровано в лістингу 2.2, який відповідає за керування змінами між сторінками. Це дозволяє забезпечити узгоджену поведінку інтерфейсу під час навігації та покращує загальне сприйняття системи користувачем.

Лістинг 2.2 – Налаштування анімації переходів

```
pageTransition: { name: "page", mode: "out-in" }
```

Кінець лістингу 2.2

Завдяки цьому механізму переміщення між екранами виглядає природно, без різких змін контенту, що створює відчуття цілісності системи. Для візуалізації таких переходів у файли стилів було додано код, який подано в лістингу 2.3.

Лістинг 2.3 – Стилі для реалізації плавних переході

```

.page-enter-active,
.page-leave-active {
  transition: all 0.4s;
}
.page-enter-from,
.page-leave-to {
  opacity: 0;
  filter: blur(8px);
}

```

Кінець лістингу 2.3

Після виконання базового налаштування проєкту було розроблено головну сторінку веб-застосунку, що містить ключові елементи інтерфейсу: банер, інформаційні картки, секцію зі статистикою, SEO-блок, а також два окремі свайпери – блок блогу та блок статей. Приклад головної сторінки продемонстровано на рисунка 2.5.

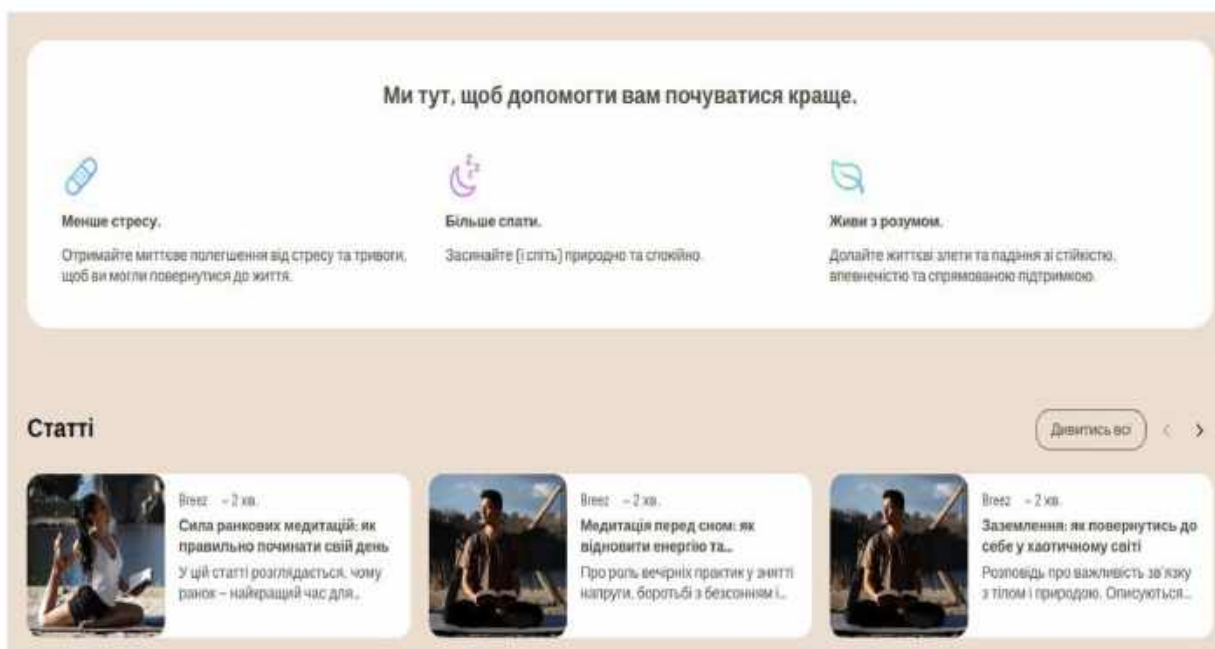


Рисунок 2.5 – Вигляд головної сторінки

Особливістю реалізації головної сторінки є використання універсального компоненту «BaseSwiper», який дає можливість створювати різні типи слайдерів

без необхідності дублювання коду. Обидва свайпери побудовані на основі цього компоненту та конфігуруються через «slot».

«slot» – це механізм передачі розмітки від батьківського компонента до дочірнього. Це дозволяє визначати структуру внутрішнього вмісту компоненту ззовні, не змінюючи його логіки.

У реалізованому компоненті «BaseSwiper» застосовано два види «slot». Перший – звичайний «slot», який використовується для виведення заголовка або будь-якого додаткового вмісту. Наприклад, у свайпері блогу через нього було передано елемент заголовка, продемонстрованого в лістингу 2.4.

Лістинг 2.4 – Приклад використання «slot»

```
<template>
  <span class="swiper__wrapper-title">Блог</span>
</template>
```

Кінець лістингу 2.4

Другий – іменованій «slide», що дає змогу передавати шаблон кожного окремого слайда. Це дозволяє формувати вміст свайпера відповідно до конкретних даних та забезпечує гнучкість у виборі типу вмісту, який має відображатися. Наприклад, для виведення карток блогу у цьому «slot» використано компонент «CardBlog», який відповідає за рендеринг окремої карточки блогу. Відповідна конструкція продемонстрована в лістингу 2.5.

Лістинг 2.5 – Приклад використання «slot»

```
<template #slide="{ item }">
  <CardBlog :data="item" />
</template>
```

Кінець лістингу 2.5

Наступним кроком було реалізовано сторінку аналітики. Ця сторінка призначена для візуалізації даних щодо емоційного стану користувача, а також для взаємодії з календарем настрою. На цій сторінці розміщено інтерактивний

календар, форму для фіксації значень показників настрою та два графіки, що демонструють динаміку змін емоційного стану. Така організація інтерфейсу дає можливість користувачу не лише фіксувати власні емоційні стани, а й порівнювати їх між собою, відстежуючи тенденції протягом певного періоду.

Для реалізації календаря було використано компонент «VueDatePicker» з підтримкою локалізації. Компонент відображається у вбудованому «inline-режимі», що дозволяє користувачеві швидко обирати дату без додаткових вікон чи модальних елементів. Це суттєво підвищує зручність використання та робить взаємодію зі сторінкою більш інтуїтивною. Відповідний фрагмент коду наведено у лістингу 2.6.

Лістинг 2.6 – Приклад використання компонента VueDatePicker

```
import VueDatePicker from "@vuepic/vue-datepicker";
import { uk } from "date-fns/locale";
const date = ref(new Date());
<VueDatePicker
  v-model="date"
  :enable-time-picker="false"
  :inline="true"
/>
```

Кінець лістингу 2.6

Календар слугує ключовою точкою взаємодії користувача із системою, оскільки вибрана дата використовується для формування та відображення відповідних аналітичних показників у графіках. Його коректна інтеграція забезпечує логічний зв'язок між інтерфейсними елементами та даними, які оновлюються в реальному часі залежно від вибору користувача.

Важливою перевагою компонента є можливість локалізації, що дозволяє адаптувати календар під мовні особливості користувача: назви місяців, днів тижня та формат дати відображаються відповідно до вибраної мови. Це позитивно впливає на зручність роботи та зменшує ризик неправильного трактування дати. Зовнішній вигляд календаря продемонстровано на рисунку 2.6.

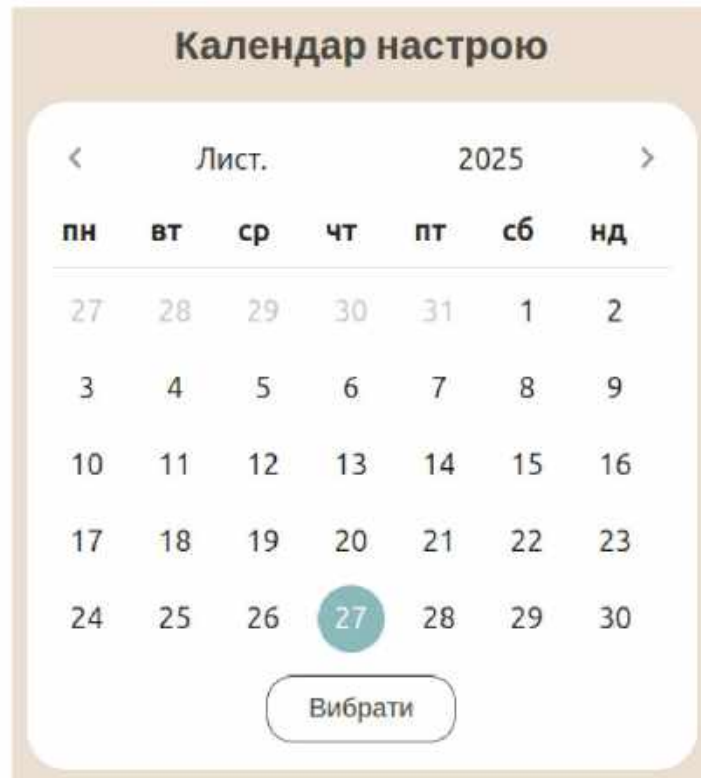


Рисунок 2.6 – Вигляд календаря

Перший графік на сторінці аналітики демонструє зміну рівня настрою та рівня тривоги протягом місяця. Для його побудови використано бібліотеку «Chart.js», що забезпечує плавну візуалізацію з підтримкою анімацій і різних режимів інтерполяції. Графік реалізовано за допомогою HTML-тега «canvas», на який ініціалізується новий екземпляр «Chart». Фрагмент коду ініціалізації наведено в лістингу 2.7.

Лістинг 2.7 – Ініціалізація лінійного графіка

```
new Chart(line.value, {
  type: "line",
  data: {
    datasets: [
      {
        label: "Рівень настрою",
        data: moodData,
        cubicInterpolationMode: "monotone",
        tension: 0.4,
        borderWidth: 3,
        pointRadius: 2,
      },
    ],
  },
});
```

```

    {
      label: "Рівень тривоги",
      data: anxietyData,
      cubicInterpolationMode: "monotone",
      tension: 0.4,
      borderWidth: 3,
      pointRadius: 2,
    },
  ],
},
);

```

Кінець лістингу 2.7

Графік має дві лінії: перша відображає зміни настрою користувача, тоді як друга демонструє рівень тривоги. Завдяки використанню інтерполяції типу «monotone» у поєднанні з параметром «tension» візуалізація набуває плавності, що робить графік більш природним для сприйняття. Такий підхід дозволяє підкреслити тренди без різких переломів, забезпечуючи краще розуміння динаміки емоційного стану в залежності від обраного періоду, приклад графіка продемонстровано на рисунку 2.7.

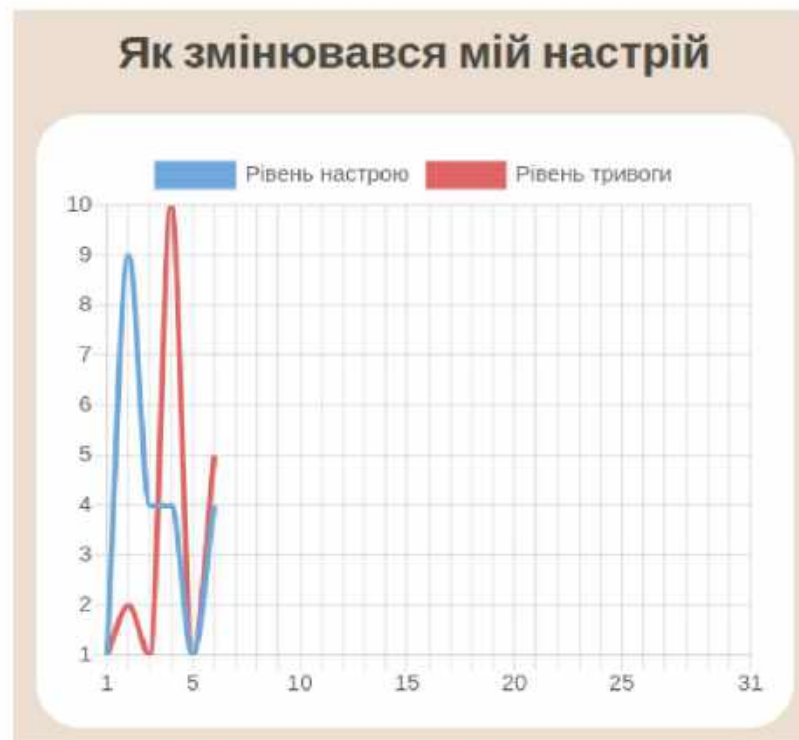


Рисунок 2.7 – Вигляд графіку

Другий графік на сторінці представлений у вигляді кругової діаграми, яка демонструє розподіл емоцій протягом заданого інтервалу. Цей компонент допомагає користувачеві швидко оцінити, які емоції домінували протягом дня чи іншого часових проміжків. Для його побудови використано бібліотеку «Chart.js», застосувавши тип «doughnut», що забезпечує стильний та компактний спосіб подання інформації. Фрагмент коду ініціалізації наведено в лістингу 2.8.

Лістинг 2.8 – Ініціалізація кругової діаграми

```

new Chart(bars.value, {
  type: "doughnut",
  data: {
    labels: emotions.value.map((e) => e.name),
    datasets: [
      {
        data: emotions.value.map((e) => e.value),
        backgroundColor: emotions.value.map((e) => e.color),
      },
    ],
  },
  options: {
    responsive: true,
    cutout: "40%",
  },
});

```

Кінець лістингу 2.8

Завдяки використанню окремого кольору для кожної емоційної категорії кругова діаграма дає змогу швидко та інтуїтивно визначити емоційний баланс користувача, а також наочно оцінити пропорції позитивних, нейтральних і негативних станів. Така форма візуалізації дозволяє з першого погляду зрозуміти загальну картину емоційного фону, визначити переважні настрої та виявити можливі дисбаланси. Це особливо корисно під час аналізу динаміки емоцій або в ситуаціях, коли необхідно оперативно зорієнтуватися в даних і з'ясувати, які стани можуть потребувати додаткової уваги чи корекції.

Впровадження кругової діаграми значно покращує користувацький досвід, оскільки забезпечує швидке та зручне сприйняття інформації без потреби опрацьовувати великі обсяги текстових або числових даних. Завдяки своїй

простоті та зрозумілості вона є ефективним інструментом для представлення результатів емоційного аналізу, що робить систему більш доступною та дружньою для користувача. Приклад графіка наведено на рисунку 2.8.



Рисунок 2.8 – Вигляд графіку

Окремим елементом інтерфейсу є система інформуючих повідомлень, що використовується для зворотного зв'язку під час збереження даних. Для цього застосовано бібліотеку «vue3-toastify». Реалізація передбачає кастомну функцію «customToast», яка дає змогу формувати повідомлення різних типів: інформаційні, успішні, попереджувальні та помилкові, приклад функції наведено в лістингу 2.9.

Лістинг 2.9 – Реалізація інформуючих повідомлень

```
export const customToast = (message: string, type: ToastType = "default")
=> {
  const options: ToastOptions = {
    position: toast.POSITION.BOTTOM_RIGHT,
    autoClose: 3000,
    hideProgressBar: true,
    transition: toast.TRANSITIONS.SLIDE,
```

```

theme: "light",
icon: ({ theme, type }) =>
  h(success, { theme, type }),
};
};

```

Кінець лістингу 2.9

Таке рішення забезпечує уніфікований стиль повідомлень у межах усього веб-застосунку та дозволяє швидко додавати нові варіанти інформування без зміни основної логіки. Результат готового повідомлення продемонстровано на рисунку 2.9.

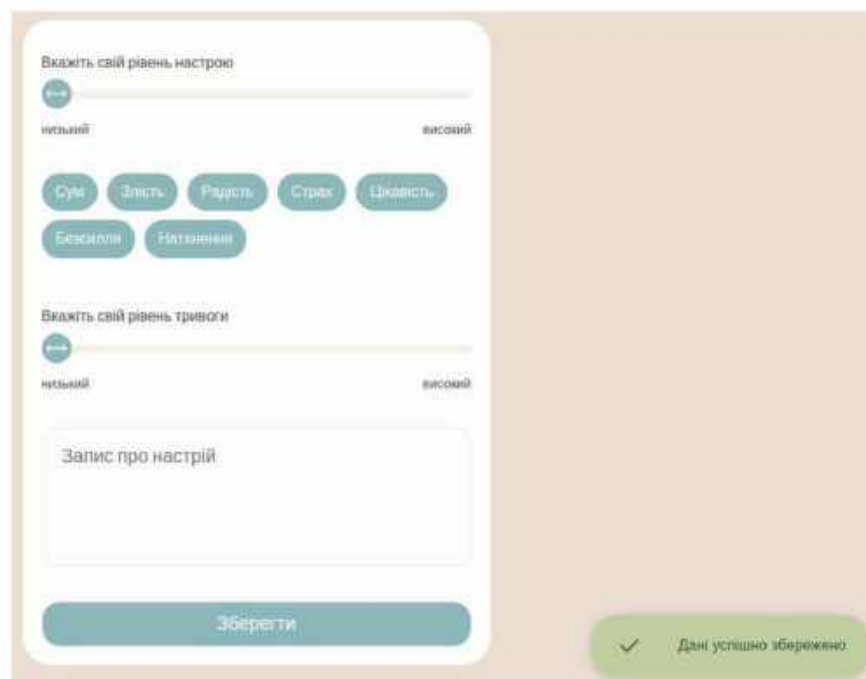


Рисунок 2.9 – Вигляд інформуючого повідомлення

Другим етапом розробки веб-застосунку було створення серверної частини, яка відповідає за автентифікацію користувача, обробку даних форм та збереження інформації у базі даних, а також надання статистики для відображення на фронтенді. Основним завданням цього етапу є організація взаємодії з базою даних та формування API для роботи клієнтської частини.

Міграції – це механізм, який дозволяє програмно створювати, змінювати та керувати структурою бази даних. Вони працюють як контроль версій для

схеми бази даних. Кожна міграція описує зміни наприклад, створення таблиці або додавання поля.

Для зберігання даних про активність та настроїв користувача були створені дві таблиці: «usertrackers» та «usertracker_emotion». Міграція таблиць наведена у лістингу 2.10.

Лістинг 2.10 – Міграція таблиць для відстеження даних користувача

```
public function up(): void
{
    Schema::create('usertrackers', function (Blueprint $table) {
        $table->uuid('id')->primary();
        $table->string('type');
        $table->string('period');
        $table->decimal('value')->default(0);
        $table->json('data')->nullable();
        $table->date('fixed_at')->nullable();
        $table->timestamps();
        $table->uuid('user_id')->constrained()->onDelete('cascade');
    });
    Schema::create('usertracker_emotion', function (Blueprint $table)
    {
        $table->uuid('usertracker_id')->index();
        $table->uuid('emotion_id')->index();
        $table->timestamp('updated_at')->nullable();
        $table->foreign('usertracker_id')->references('id')->
        >on('usertrackers')->onDelete('cascade');
        $table->foreign('emotion_id')->references('id')->on('items')->
        >onDelete('cascade');
    });
}
```

Кінець лістингу 2.10

У таблиці «usertrackers» зберігаються основні дані про активність користувача, включаючи тип запису, період, значення та додаткові дані у форматі JSON. Таблиця «usertracker_emotion» забезпечує зв'язок між записами користувача та емоціями для аналітики.

Контролер – це компонент серверної частини веб-застосунку, який відповідає за обробку запитів від клієнта та формування відповідей. Він отримує

дані, передані з форми чи API-запиту, викликає необхідну бізнес-логіку, взаємодіє з моделями та повертає результат у вигляді JSON, HTML або іншого формату.

Для створення нового користувача реалізовано контролер «RegisterController». Контролер обробляє запит реєстрації, валідуючи дані через «RegisterRequest» і зберігаючи користувача у базі даних. Приклад коду контролера продемонстровано у лістингу 2.11.

Лістинг 2.11 – Контролер для реєстрації користувача

```
public function register(RegisterRequest $request)
{
    $user = StoreUserAction::run($request->validated());
    return response()->json([
        'message' => 'Вітаємо! Реєстрація успішна',
        'state' => 'active',
    ] + $this->authResource($user));
}
```

Кінець лістингу 2.11

Валідація даних при реєстрації виконується через «RegisterRequest», що перевіряє унікальність «email» та телефону, формат паролю та інші обов'язкові поля.

Валідація – це процес перевірки даних, які надходять від користувача, на відповідність визначеним правилам перед їх обробкою або збереженням у базі даних. Її основна мета – гарантувати, що введена інформація є коректною, повною та безпечною.

Для збереження даних форми настрою реалізовано метод «tracks» у контролері «UsertrackerController». Він приймає дані від користувача та зберігає їх у таблиці «usertrackers», а також синхронізує зв'язок з емоціями. Приклад коду для методу продемонстровано у лістингу 2.12.

Лістинг 2.12 – Метод для збереження даних настрою

```
public function tracks(UsertrackerRequest $request){
    $user = $request->user();
```

```

        $data = $request->getData();
        $usertracker = $user->usertracks()->create(['data' => $data]);
        if ($emotionIds = $request->input('emotions', [])) {
            $usertracker->emotions()->sync(Arr::pluck($emotionIds,
'id'));
        }
        return response()->json(['message' => 'Дані успішно збережено!',
'id' => $usertracker->id]);
    }

```

Кінець лістингу 2.12

Маршрути – це правила, які визначають, на якій URL-адресі та яким методом (GET, POST, PUT, DELETE) можна звернутися до певної функції або контролера на сервері. Вони забезпечують чітку організацію взаємодії між клієнтом і бекендом, дозволяючи передавати запити у відповідні частини логіки застосунку. Крім того, маршрути допомагають структурувати API, роблячи його передбачуваним і зрозумілим: кожен шлях відповідає за конкретну дію отримання даних, створення нового запису, оновлення інформації або її видалення. Завдяки маршрутам система може контролювати доступ, застосовувати «middleware» та реалізовувати безпечну й логічно упорядковану обробку запитів.

Для роботи з бекендом були створені маршрути для реєстрації та роботи з треками користувача. Приклад коду наведено у лістингу 2.13 та лістингу 2.14.

Лістинг 2.13 – Маршрут реєстрації

```

Route::post('register',
[\App\Http\Auth\Api\Controllers\RegisterController::class, 'register']);

```

Кінець лістингу 2.13

Лістинг 2.14 – Маршрути для роботи з треками користувачами

```

Route::group(['middleware' => 'auth:sanctum', 'prefix' => 'my'], function
() {
    Route::group(['prefix' => 'tracks'], function () {
        Route::post('/',
[\App\Http\Client\Api\Controllers\My\UsertrackerController::class,
'tracks']);
    });
}

```

```

    Route::get('/statistic',
[\App\Http\Client\Api\Controllers\My\UsertrackerController::class,
'statistic']);
    });
});

```

Кінець лістингу 2.14

Третім етапом розробки веб-застосунку було підключення API та реалізація механізмів взаємодії фронтенду із серверною частиною. Основним завданням є отримання даних для відображення на головній сторінці, а також організація автентифікації користувача.

Для отримання даних із серверної частини застосунку в Nuxt 3 використовується зв'язка «useAsyncData» та «\$fetch». Функція «useAsyncData» дає змогу завантажувати дані асинхронно як на сервері, так і на клієнті, автоматично підключаючи їх до реактивної системи Vue. Вона також керує станами завантаження «pending», обробкою помилок «error» і дозволяє виконувати попередню трансформацію результату перед тим, як дані будуть передані компоненту.

Функція «\$fetch» є зручнішою обгорткою над стандартним «Fetch API». Вона спрощує взаємодію з «REST-ендпоінтами», автоматично обробляє JSON-відповіді, підтримує роботу з заголовками, авторизацією та коректно взаємодіє з механізмами «SSR» у Nuxt 3.

Для централізованого керування параметрами HTTP-запитів було створено функцію «defaultOptions», яка формує базову конфігурацію, включаючи доступний токен, стандартні заголовки та інші необхідні параметри. Її реалізацію продемонстровано в лістингу 2.15.

Лістинг 2.15 – Функція для формування стандартних параметрів

```

export function defaultOptions<T>() {
  const config = useRuntimeConfig();
  const access_token = useCookie("access_token", { maxAge:60*60*24*30});
  const rawHeaders = useRequestHeaders();
  const defaults: UseFetchOptions<T> = {
    baseUrl: config.public.API_BASE_URL || "",

```

```

headers: {
  Accept: "application/json",
  "Cache-Control": "no-cache",
  Authorization: `Bearer ${access_token.value || ""}`,
  sHost: "Breez",
},
};
return defaults;
}

```

Кінець лістингу 2.15

Для завантаження даних головної сторінки було застосовано асинхронний запит. Цей запит виконується під час рендерингу сторінки та забезпечує отримання усієї необхідної інформації з бекенд-сервера. Завдяки використанню вищезгаданих функцій та методів, процес отримання даних відбувається у реактивному режимі, що дозволяє оновлювати інтерфейс без додаткових викликів перезавантаження сторінки. Реалізація запита наведена у лістингу 2.16.

Лістинг 2.16 – Запит для отримання даних головної сторінки

```

const { data: page } = await useAsyncData(
  `home`,
  () =>
    $fetch(`/pages/home`, {
      ...defaultOptions(),
    }),
  {
    transform: (page) => {
      let blocksObj = {};
      page.blocks.forEach((el) => {
        blocksObj[el.type] = el;
      });
      return { ...blocksObj, data: page.data };
    },
  }
);

```

Кінець лістингу 2.16

У цьому прикладі параметр «transform» використовується для попередньої обробки отриманих даних: з масиву блоків створюється об'єкт «blocksObj», що значно спрощує доступ до потрібного блоку без додаткових пошуків. Це покращує читабельність коду та підвищує продуктивність при роботі з великими

наборами даних. Результат завантаження даних головної сторінки продемонстровано на рисунку 2.10.

```

index.vue:34
Proxy(Object) (hero: {...}, support_section: {...}, posts: {...}, harmony_tech: {...},
meditations: {...}, ...)
  [[Handler]]: MutableReactiveHandler
  [[Target]]: Object
    data: {id: '019a3ab8-b6b3-7162-bfb0-2365b1f702c6', entity: 'page', type: 'p...
    harmony_tech: {id: 4, name: '*** Технології для гармонійного життя (Головна)
    hero: {id: 1, name: '*** Верхній блок (Головна) - НАПОВНІТЬ БЛОК ***', slug
    meditations: {id: 5, name: '*** Медитації (Головна) - НАПОВНІТЬ БЛОК ***',
    posts: {id: 3, name: '*** Статті (Головна) - НАПОВНІТЬ БЛОК ***', slug: 'ro...
    support_section: {id: 2, name: '*** Ми тут, щоб допомогти почуватися вам кр...
    [[Prototype]]: Object
    [[IsRevoked]]: false
  
```

Рисунок 2.10 – Результат отримання даних головної сторінки

Для автентифікації користувача реалізовано функцію «onSubmit», яка відправляє POST-запит на «endpoint» «/login» і обробляє отриманий токен доступу. Приклад функції продемонстровано в лістингу 2.17.

Лістинг 2.17 – Функція для авторизації користувача

```

const onSubmit = async (val, action) => {
  try {
    loader.value = true;
    const data = await $fetch("/login",
      {
        ...defaultOptions(),
        method: "POST",
        body: form.value,
      }
    );
    modalStore.toggleModal("login", false);
    access_token.value = data.token;
    if (route.path === "/login")
      {
        navigateTo({ path: "/profile" }, { redirectCode: 301 });
      }
    else (route.path !== "/login") {
      customToast("Ви успішно увійшли у свій акаунт", "success");
    }
  } catch (error) {
    action.setErrors(error.data.errors);
  } finally {
  
```

```

    loader.value = false;
  }
};

```

Кінець лістингу 2.17

Функція забезпечує відображення індикатора завантаження `loader` під час обробки операції входу, що дозволяє користувачу бачити, що запит виконується. Після отримання відповіді від сервера токен доступу зберігається у `cookie` або іншому сховищі, що дає змогу надалі працювати з приватними API-маршрутами.

Логіка подальшої взаємодії залежить від того, яким чином була відкрита форма авторизації. Якщо вхід здійснювався через модальне вікно, то після успішної авторизації це вікно автоматично закривається, а користувач отримує сповіщення про успішний вхід, при цьому перенаправлення не виконується і користувач залишається на поточній сторінці. У випадку, коли форма авторизації була відкрита як окрема сторінка «`/login`», після успішного отримання токена система виконує перенаправлення на приватний маршрут «`/profile`», де користувач може працювати зі своїми персональними даними. Результат успішної авторизації зображено на рисунку 2.11.

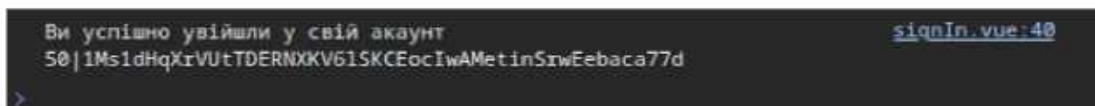


Рисунок 2.11 – Результат успішної авторизації користувача

У разі якщо користувач вводить некоректні дані для входу, сервер повертає помилку, яка обробляється в блоці «`catch`» в функції «`onSubmit`». Також у форму передаються отримані від сервера повідомлення про невірні поля, що дає можливість коректно підсвітити та пояснити помилки вводу. Такий підхід підвищує зручність використання форми авторизації та забезпечує швидке розуміння причини відмови у вході. Вигляд повідомлення про помилку наведено на рисунку 2.12.

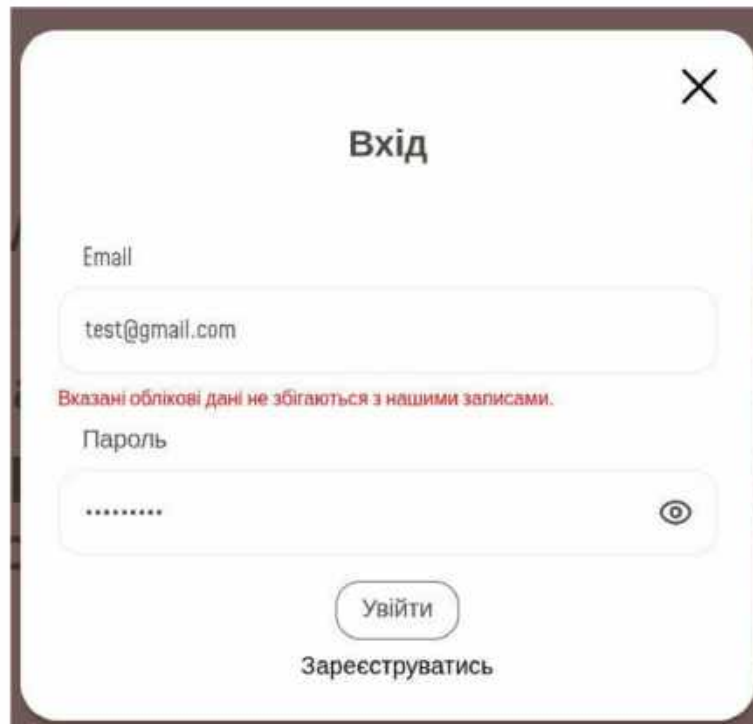


Рисунок 2.12 – Вигляд інформуючого повідомлення про помилку авторизації

«Middleware» – це проміжний програмний шар, який виконується перед обробкою запиту або перед переходом на певний маршрут у веб-застосунку. Його основне призначення – перехоплювати навігацію чи запити та виконувати додаткову логіку, таку як перевірка автентифікації або перенаправлення користувача.

Для контролю доступу до приватних сторінок було створено «middleware», який перевіряє наявність токена доступу перед переходом на маршрути «/profile» та «/analytics». Якщо токен відсутній, користувач автоматично перенаправляється на сторінку авторизації «/login»). Також реалізована логіка, що запобігає доступу авторизованих користувачів до сторінки входу, перенаправляючи їх на головну сторінку. Приклад функції продемонстровано в лістингу 2.18.

Лістинг 2.18 – Middleware для контролю доступу до приватних сторінок

```
export default defineNuxtRouteMiddleware((to, from) => {
  const access_token = useCookie('access_token');
  if (to.path.includes('/profile') && !access_token.value) {
```

```
    return navigateTo('/login');
  }
  if (to.path.includes('/analytics') && !access_token.value) {
    return navigateTo('/login');
  }
  if (to.path.includes('/login') && access_token.value) {
    return navigateTo('/');
  }
});
```

Кінець лістингу 2.18

Завдяки такому підходу забезпечується централізований контроль доступу до конфіденційних даних. «Middleware» гарантує, що приватні сторінки будуть доступні лише автентифікованим користувачам, підвищуючи безпеку застосунку та спрощуючи управління логікою доступу під час масштабування проекту.

РОЗДІЛ 3

ЕКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ ТЕХНІЧНИХ ХАРАКТЕРИСТИК ВЕБ-ЗАСТОСУНКУ ДЛЯ СТЕЖЕННЯ ЗА МЕНТАЛЬНИМ ЗДОРОВ'ЯМ

3.1 Методика проведення дослідження

Після завершення процесу розробки веб-застосунку для стеження за ментальним здоров'ям розпочинається етап експериментального дослідження його технічних характеристик. Метою цього етапу є визначення рівня ефективності, продуктивності, стабільності та відповідності сучасним вимогам веб-розробки. Отримані результати дають змогу оцінити готовність системи до подальшого розгортання, масштабування та використання реальними користувачами.

Оскільки аналіз виконувався у локальному середовищі, усі вимірювання проводилися без впливу зовнішніх мережевих факторів та без застосування онлайн-сервісів аналітики. Основними інструментами дослідження стали «Google Lighthouse» та «Chrome DevTools», які забезпечують комплексний огляд технічного стану сучасних веб-застосунків.

«Google Lighthouse» це автоматизований інструмент аудиту якості веб-додатків, інтегрований у браузер «Google Chrome». Він аналізує застосунок за ключовими напрямками:

- продуктивність – оцінює швидкість завантаження сторінки, час рендерингу ключових елементів, ефективність роботи скриптів і загальну реактивність інтерфейсу;
- доступність – перевіряє коректність контрастності, наявність альтернативних текстів, структурованість розмітки та зручність навігації для користувачів з особливими потребами;
- SEO – визначає технічну готовність застосунку до індексації та відповідність критеріям видимості для пошукових систем.

Для забезпечення розширеного технічного аналізу роботи інтерфейсу використовувався «Chrome DevTools». За його допомогою досліджувалися час рендерингу компонентів, кількість перерисовок інтерфейсу, навантаження на основний потік, наявність блокуючих ресурсів, плавність анімацій, а також коректність адаптивного відображення сторінок на різних розширеннях екрана. Сукупність цих показників дозволила детально простежити поведінку застосунка у реальних сценаріях взаємодії.

У межах експериментального дослідження особливу увагу було приділено «Core Web Vitals» – стандартизованим метрикам, що відображають реальний користувацький досвід. Показник «LCP» (Largest Contentful Paint) демонструє, як швидко з'являється найбільший видимий елемент сторінки; оптимальним є значення до 2,5 секунди. Показник «CLS» (Cumulative Layout Shift) характеризує стабільність макета під час завантаження: мінімальні зсуви елементів забезпечують комфортний UX, тому рекомендоване значення становить менше 0,4. Показник «FCP» (First Contentful Paint) визначає момент появи першого видимого елемента, що дозволяє оцінити початкову швидкість реакції інтерфейсу; оптимальний час – до 1,8 секунди.

3.2 Обробка та аналіз отриманих результатів

Після проведення всіх етапів тестування було отримано комплекс технічних та поведінкових показників, які дали можливість оцінити результативність роботи веб-застосунку для стеження за ментальним здоров'ям. Дані були згруповані за ключовими напрямками: продуктивність, валідність коду, адаптивність, доступність, поведінка користувачів та навантажувальна стійкість системи.

Експериментальне дослідження технічних характеристик веб-застосунку для стеження за ментальним здоров'ям передбачає детальне вимірювання основних веб-показників, аналіз продуктивності, доступності, SEO та дотримання сучасних веб-стандартів. На основі отриманих результатів

проводиться комплексна інтерпретація, порівняння з рекомендованими нормами Google та формування висновків щодо якості розробленого застосунка.

Оцінювання проводилося у локальному середовищі з використанням інструментів «Google Lighthouse» та «Chrome DevTools». Це дало можливість отримати точні технічні результати без впливу зовнішніх мережових факторів, а також забезпечило повторюваність експерименту. Дані, отримані за допомогою «Google Lighthouse», були згруповані за напрямками оцінювання:

- продуктивність;
- доступність;
- технічна оптимізація.

Кожен із цих параметрів оцінюється за шкалою від 0 до 100, де 90-100 відповідає максимальному рівню якості. Для зручності аналізу дані були презентовані у табличному та графічному вигляді. Рекомендовані значення взяті відповідно до офіційних стандартів Google. Тестування продуктивності дозволило визначити загальну швидкість завантаження застосунка, час взаємодії та оптимізованість ресурсів. Основні показники продуктивності представлені в таблиці 3.1.

Таблиця 3.1 – Показники продуктивності «Google Lighthouse»

Показник	Значення	Рекомендовано	Висновок
Performance (загальний бал)	82	≥ 90	Задовільний результат
First Contentful Paint (FCP)	1.2 s	≤ 1.8 s	У нормі
Largest Contentful Paint (LCP)	2.0 s	≤ 2.5 s	У нормі
Speed Index	2.8 s	≤ 3.4 s	У нормі
Time to Interactive (TTI)	$\approx 2.0-2.5$ s	≤ 3.8 s	У нормі
Total Blocking Time (TBT)	20 ms	≤ 200 ms	Відмінно

Отримані результати показують, що основний контент відображається достатньо швидко, а взаємодія з інтерфейсом доступна практично одразу після завантаження сторінки. Низький показник «ТБТ» свідчить про якісну роботу

JavaScript-процесів та відсутність блокуючих скриптів, що могли б затримувати «рендеринг». Загалом застосунок демонструє стабільну продуктивність, а всі ключові метрики перебувають у межах рекомендованих значень.

Доступність є важливою характеристикою сучасних веб-застосунків, особливо у сфері ментального здоров'я, де інтерфейс має бути зрозумілим, інклюзивним та комфортним для користувача. Перевірка за допомогою «Google Lighthouse» показала повну відповідність застосунка сучасним вимогам щодо доступності. Усі базові критерії були виконані, що підтверджує якісну реалізацію інтерфейсу. Основні показники доступності представлені в таблиці 3.2.

Таблиця 3.2 – Показники доступності

Параметр	Значення	Рівень відповідності	Висновок
Accessibility (загальний бал)	100	≥ 90	Відмінний результат
Контрастність тексту	Нормальна	AA	Контраст відповідає стандартам
Alt-тексти	100%	Повна наявність	Усі зображення мають коректні альтернативні описи
Структура заголовків	Семантична	Відповідає	Логічне та послідовне використання тегів

Отримані результати показують, що основний контент відображається достатньо швидко, а взаємодія з інтерфейсом доступна практично одразу після завантаження сторінки. Низький показник «ТВТ» свідчить про ефективну роботу JavaScript-процесів та відсутність блокуючих скриптів.

Хоча застосунок тестувався локально, «Google Lighthouse» дозволив оцінити його технічну готовність до пошукової оптимізації. Результати вказують на високий рівень відповідності сучасним вимогам SEO, що свідчить про якісну

структуру сторінки та правильну конфігурацію основних тегів. Основні SEO-показники представлені в таблиці 3.3.

Таблиця 3.3 – SEO-показники

Параметр	Значення	Рівень відповідності	Висновок
SEO (загальний бал)	100	≥ 90	Відмінний результат
Наявність title	Є	Обов'язково	Тег «title» заповнений коректно
Параметр	Значення	Рівень відповідності	Висновок
Наявність «meta description»	Є	Бажано	Мета-опис присутній та оптимізований
Семантична розмітка	Правильна	Рекомендовано	Логічна структура HTML

Отримані результати свідчать, що застосунок технічно готовий до розміщення у відкритому доступі. Структура HTML оптимізована, семантичні елементи реалізовані коректно, а сторінка відповідає вимогам адаптивності, доступності та SEO-індексації. Це забезпечує не лише комфорт користувача, а й позитивний вплив на видимість застосунка у пошукових системах після його розгортання.

ВИСНОВКИ

У рамках кваліфікаційної роботи магістра було успішно розроблено та досліджено веб-застосунок для моніторингу ментального здоров'я користувача. Проведена робота дозволила досягти поставленої мети – створення інтерактивної системи, яка забезпечує зручний доступ до особистих показників психологічного стану та дозволяє ефективно відстежувати зміни емоційного стану у часі.

У процесі виконання роботи були повністю реалізовані всі поставлені завдання:

- здійснено огляд існуючих веб-рішень для підтримки ментального здоров'я. Проведено комплексний огляд існуючих цифрових сервісів, визначено їхні ключові переваги та недоліки. Проаналізовано сучасні наукові підходи та методи створення веб-застосунків у цій сфері, що стало теоретичною основою для проектування системи;

- обрано сучасні інструменти та технології для реалізації клієнтської та серверної частини веб-застосунку. Обґрунтовано вибір технологічного стеку, що забезпечує високу продуктивність, надійність та масштабованість. Для реалізації клієнтської частини обрано фреймворк Nuxt 3 на базі Vue.js та SCSS для стилізації, а для серверної частини – фреймворк Laravel на базі PHP;

- розроблено фронтенд-частину веб-застосунку для взаємодії користувача з системою. Створено функціональну фронтенд-частину із розробленою структурою та логікою навігації, що забезпечує швидкий доступ до основних функцій ведення щоденника, управління профілем та перегляд аналітики. Забезпечено інтуїтивний користувацький інтерфейс та ефективну візуалізацію динаміки емоційних показників;

- розроблено структуру бази даних для збереження показників ментального стану користувача. Спроектовано та реалізовано структуру бази даних, яка забезпечує надійне збереження конфіденційної інформації про

ментальний стан користувача, гарантуючи цілісність даних та їхню готовність до аналітичної обробки;

– реалізовано підключення та інтеграцію API між клієнтською та серверною частинами веб-застосунку. Успішно реалізовано підключення та інтеграцію API для обміну даними. Впроваджено необхідні механізми автентифікації та авторизації, що забезпечило централізований та безпечний контроль доступу до приватних сторінок та конфіденційних даних;

– проведено тестування працездатності веб-застосунку. Проведено експериментальну оцінку ефективності розробленого застосунку, включаючи перевірку технічних характеристик. Виконано експериментальну оцінку ефективності та функціональності розробленого застосунку. Результати тестування, зокрема з використанням «Google Lighthouse», підтвердили стабільну роботу системи, високу продуктивність та повну відповідність сучасним вимогам щодо доступності;

Результатом виконаної роботи є повністю функціональний веб-застосунок, який дозволяє користувачеві здійснювати самооцінку та самоспостереження, зберігати дані про психологічний стан та аналізувати їх у часовому вимірі. Наукова новизна роботи полягає у запропонованому підході до інтеграції механізмів самооцінки, візуального аналізу емоційних показників та зручного інтерфейсу веб-застосунку.

Практична цінність роботи підтверджується можливістю використання прототипу для особистого контролю психологічного стану, а також у навчальних, корпоративних та медичних середовищах. Отримані результати відкривають перспективи для подальшого розвитку системи, зокрема шляхом впровадження алгоритмів машинного навчання для прогнозування емоційних станів, розширення функціоналу та інтеграції з мобільними платформами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гергель В. В., Кошелюк В. А. Використання веб-технологій у діагностиці та профілактиці ментальних розладів. Науковий журнал «Комп'ютерно-інтегровані технології: освіта, наука, виробництво». Луцьк: Луцький НТУ, 2025. Вип. № 59. С.76-81.
2. Сверстюк А. С., Андрущак І. Є., Гергель В. В. Безпека та конфіденційність даних у веб-застосунках для ментального здоров'я: ризики та методи захисту. Науковий журнал «Студентський науковий вісник». Луцьк: Луцький НТУ, 2025. Вип. № 54. С.141-147.
3. Linardon J., Cuijpers P., Carlbring P., Messer M., Fuller-Tyszkiewicz M. Genetic evidence suggests a genetic association between major depressive disorder and reduced cortical gray matter volume. A Mendelian randomization study and mediation analysis - ScienceDirect. URL: <https://surli.cc/nmfziz> (дата звернення: 05.09.2025).
4. Kurapov A., Danyliuk I., Loboda A., Kalaitzaki A., Kowatsch T., Klimash T., Predko V. Six months into the war: a first-wave study of stress, anxiety, and depression among in Ukraine. URL: <https://surl.lu/cbjbfy> (дата звернення: 06.09.2025).
5. Офіційний веб-сайт «hedepy». URL: <https://hedepy.com.ua> (дата звернення: 08.09.2025).
6. Офіційний веб-сайт «ТИ ЯК?». URL: <https://howareu.com/> (дата звернення: 08.09.2025).
7. Офіційний веб-сайт «mentalcode». URL: <https://mentalcode.com.ua/> (дата звернення: 08.09.2025).
8. World mental health report: transforming mental health for all. URL: <https://surl.li/wvkvbp> (дата звернення: 09.09.2025).
9. Science Direct. A digital self-help tool to promote mental well-being for Ukrainians affected by war – Assessing predictors of stress. URL: <https://surl.li/mokkso> (дата звернення: 10.09.2025).

10. Iwaya L., Babar M., Rashid A., Wijayarathna G. On the privacy of mental health apps: An empirical investigation and its implications for app development. JMIR Mental Health. URL: <https://surl.li/tkmlju> (дата звернення: 12.09.2025).
11. Психологічна підтримка в умовах соціальної нестабільності. URL: <https://surl.li/zsqqqf> (дата звернення: 14.09.2025).
12. Vue.js Documentation. The Progressive JavaScript Framework. URL: <https://vuejs.org/> (дата звернення: 16.09.2025).
13. Nuxt 3 Documentation. The intuitive Vue framework for building full-stack web applications. URL: <https://nuxt.com/> (дата звернення: 16.09.2025).
14. React Documentation. A JavaScript library for building user interfaces. URL: <https://react.dev/> (дата звернення: 17.09.2025).
15. Next.js Documentation. The React Framework for Production. URL: <https://nextjs.org/> (дата звернення: 17.09.2025).
16. Angular Documentation. Develop across all platforms with Angular. URL: <https://angular.io/> (дата звернення: 19.09.2025).
17. Less Documentation. Less: Leaner Style Sheets. URL: <https://lesscss.org/> (дата звернення: 23.09.2025).
18. Tailwind CSS Documentation. A utility-first CSS framework for rapid UI development. URL: <https://tailwindcss.com/> (дата звернення: 23.09.2025).
19. SCSS/Sass Official Guide. Sass: Syntactically Awesome Style Sheets. URL: <https://sass-lang.com/> (дата звернення: 26.09.2025).
20. Laravel Documentation. The PHP Framework for Web Artisans. URL: <https://laravel.com/> (дата звернення: 27.09.2025).
21. Django Documentation. The Web Framework for Perfectionists with Deadlines. URL: <https://www.djangoproject.com/> (дата звернення: 27.09.2025).
22. Django REST Framework. Powerful and flexible toolkit for building Web APIs. URL: <https://www.django-rest-framework.org/> (дата звернення: 28.09.2025).