Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ ОЦІНЮВАННЯ ПСИХОЕМОЦІЙНОГО СТАНУ КОРИСТУВАЧІВ ІЗ ВИКОРИСТАННЯМ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-22-1
Галушко А. О.
(прізвище, ініціали)

Спеціальність F3 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник ст.викл. Путятіна О. Є.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність F3 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____202
____р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Галушко Ангеліні Олексіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення вебзастосунку для оцінювання психоемоційного стану користувачів із використанням методів штучного інтелекту

затверджена наказом університету від 26 травня 2026 року № 435Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 02 липня 2026 р.
3. Вихідні дані до роботи наукові статті та технічна документація, матеріали конференцій, дані інтернет-мережі, валідовані психометричні методики оцінювання стресу, тривожності, емоційного вигорання та психологічного благополуччя, велика мовна модель Llama 3.3-70b через Groq API, мова програмування TypeScript, бібліотека React, середовище Node.js, фреймворк Express, система керування базами даних PostgreSQL, ORM-інструмент Prisma.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз сучасного стану проблеми оцінки психоемоційного стану користувачів та огляд існуючих вебзастосунків і платформ для моніторингу стресу, тривожності та емоційного вигорання.
2. Дослідження психометричних методик та можливостей застосування штучного інтелекту для аналізу результатів тестування.
3.Проєктування архітектури вебзастосунку, вибір інструментальних засобів розроблення та реалізація основних функціональних модулів системи
4.Тестування та аналіз результатів роботи системи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри): актуальність проблеми розроблення вебзастосунку для оцінки психоемоційного стану користувачів, постановка задачі, загальна архітектурна схема системи, компонентна модель інтерфейсу, схема шарової архітектури бекенду, ER-діаграма бази даних, діаграми процесів тестування та ШІ-аналізу, скриншоти інтерфейсу вебзастосунку, результати тестування.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	13.04.2026	
2	Аналіз завдання, підбір літератури	14.04.26-16.04.26	
3	Аналіз літератури з проблеми, що вирішується	17.04.26-19.04.26	
5	Формування кроків вирішення проблеми	23.04.26-05.05.26	
6	Програмна реалізація	26.04.26-10.05.26	
7	Аналіз отриманих результатів	11.05.26-14.05.26	
8	Оформлення пояснювальної записки	15.05.26-31.05.26	
9	Перевірка на нормоконтроль	01.06.26-17.06.26	
10	Перевірка на плагіат	18.06.26-30.06.26	
11	Рецензування	20.06.26-30.06.26	
12	Підготовка презентації та доповіді	20.06.26-30.06.26	
13	Занесення роботи в електронний архів	30.06.26-09.07.26	
14	Попередній захист кваліфікаційної роботи	30.06.26-09.07.26	

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ ст. викл. Путятіна О. Є.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ /ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 95 с., 9 табл., 18 рис., 5 дод., 25 джерел.

ВЕБЗАСТОСУНОК, МОНІТОРИНГ СТРЕСУ, ПЕРСОНАЛІЗОВАНІ РЕКОМЕНДАЦІЇ, ПСИХОЕМОЦІЙНИЙ СТАН, ШТУЧНИЙ ІНТЕЛЕКТ, NLP, NODE.JS, POSTGRESQL, REACT.

Об’єктом роботи є процес розроблення клієнт-серверного вебзастосунку для діагностики та моніторингу психоемоційного стану користувачів.

Метою роботи є створення системи, що автоматизує проходження психометричних тестів, аналізує відповіді за допомогою Llama 3.3-70b та формує персоналізовані рекомендації.

У роботі використано методи NLP, сентимент-аналізу, зваженого скорингу, REST API, архітектуру React/TypeScript + Node.js/Express, PostgreSQL/Prisma ORM, промпт-інжиніринг та Docker.

Практична цінність полягає у розробленні платформи з ШІ-модулем, контролем доступу, резервною генерацією порад та захистом даних.

Розроблено й протестовано систему для анонімного тестування, відстеження динаміки стану та отримання адаптованих рекомендацій.

Область застосування – корпоративні програми підтримки ментального здоров’я, освітні заклади та самоспостереження.

WEB APPLICATION, STRESS MONITORING, PSYCHO-EMOTIONAL STATE, PERSONALIZED RECOMMENDATIONS, ARTIFICIAL INTELLIGENCE, NLP, NODE.JS, POSTGRESQL, REACT.

The object of work is the development of a client-server web application for diagnosing and monitoring users’ psycho-emotional state.

The goal is to create a system that automates psychometric testing, analyzes responses using Llama 3.3-70b, and generates personalized recommendations.

The work uses NLP, sentiment analysis, weighted scoring, REST API, React/TypeScript + Node.js/Express architecture, PostgreSQL/Prisma ORM, prompt engineering, and Docker.

The practical value lies in developing a platform with an AI module, access control, fallback recommendations, and data protection.

A system for anonymous testing, emotional state tracking, and adaptive recommendation generation has been developed and tested.

The application area includes corporate mental health programs, educational institutions, and self-monitoring tools.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект

API – Application Programming Interface (інтерфейс програмування застосунків, набір протоколів для взаємодії між клієнтською та серверною частинами системи)

bcrypt – алгоритм хешування паролів із використанням випадкової «солі» та ітерацій для підвищення криптостійкості

CORS – Cross-Origin Resource Sharing (механізм контролю доступу до ресурсів з іншого домену, що забезпечує безпеку вебзастосунків)

Docker – платформа контейнеризації для ізольованого розгортання та управління сервісами застосунку

Helmet – бібліотека middleware для Node.js, що автоматично встановлює безпечні HTTP-заголовки

JWT – JSON Web Token (стандарт відкритого формату для безпечної передачі даних між сторонами у вигляді об’єкта JSON, використовується для аутентифікації)

NLP – Natural Language Processing (обробка природної мови, напрям штучного інтелекту для аналізу та інтерпретації текстової інформації)

ORM – Object-Relational Mapping (об’єктно-реляційне відображення, технологія зв’язування реляційної бази даних з об’єктно-орієнтованим кодом)

Prisma – сучасний ORM-інструмент для TypeScript, що забезпечує типобезпечну роботу з реляційною базою даних

REST API – Representational State Transfer API (архітектурний стиль побудови мережевих інтерфейсів на основі HTTP-методів)

Vite – інструмент збирання та швидкої розробки frontend-застосунків на базі нативних ES-модулів

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій та зростання темпу життя проблеми збереження ментального здоров’я набувають особливої актуальності. За даними Всесвітньої організації охорони здоров’я, рівень стресу, тривожності та емоційного вигорання серед працівників організацій та учнів освітніх закладів демонструє стійку тенденцію до зростання. Традиційні методи психологічного моніторингу, що базуються на паперових опитувальниках або індивідуальних консультаціях, характеризуються обмеженою масштабованістю, високими тимчасовими витратами та відсутністю можливості оперативного відстеження динаміки стану в реальному часі. Це зумовлює необхідність розроблення автоматизованих цифрових інструментів, здатних забезпечити своєчасне виявлення груп ризику та надання превентивної підтримки без географічних чи адміністративних обмежень.

Аналіз існуючих цифрових платформ для психологічної допомоги свідчить про наявність суттєвих прогалин у їх функціональному наповненні. Більшість комерційних рішень орієнтовані переважно на агрегацію статичних тестів без подальшого інтелектуального опрацювання результатів, або ж використовують спрощені алгоритми інтерпретації, що не враховують індивідуальний контекст та динаміку змін стану користувача. Крім того, багато систем не відповідають сучасним вимогам інформаційної безпеки, не передбачають багаторівневої архітектури контролю доступу або ігнорують етичні аспекти автоматизованої психологічної діагностики, що може призвести до некоректної інтерпретації результатів. Вирішення зазначеної проблеми потребує створення цілісної інформаційної системи, яка поєднує валідовані психометричні методики з можливостями сучасних технологій штучного інтелекту.

Стрімкий розвиток великих мовних моделей та методів обробки природної мови відкриває нові перспективи для автоматизованого аналізу психоемоційних даних. Інтеграція генеративних моделей у процес інтерпретації результатів тестування дозволяє формувати структуровані, персоналізовані рекомендації, адаптовані до конкретного рівня стресу, тривожності чи емоційного виснаження респондента. За умови ретельного налаштування системних промптів, впровадження етичних фільтрів та механізмів резервної генерації у разі недоступності сервісів, штучний інтелект здатний забезпечити високу якість семантичного аналізу без порушення принципів академічної та медичної етики. Поєднання таких підходів із сучасними вебтехнологіями створює фундамент для розроблення масштабованих клієнт-серверних систем дистанційного моніторингу благополуччя.

Обґрунтування необхідності розроблення запропонованого вебзастосунку базується на аналізі сучасного стану галузі цифрової психології та виявленні потреби в інструменті, що забезпечує повний цикл взаємодії: від реєстрації та проходження тестів до інтелектуального аналізу результатів, візуалізації аналітики та надання адаптованих порад. Особливу увагу приділено архітектурним рішенням, що гарантують конфіденційність даних, анонімність опитувань, багаторівневу авторизацію та відповідність стандартам інформаційної безпеки. Основні проєктні рішення спрямовані на створення модульної системи, здатної інтегрувати різні ШІ-провайдери, підтримувати динамічне налаштування тестових наборів та масштабуватися для використання в корпоративному або освітньому середовищі.

Тема кваліфікаційної роботи: «Розробка вебзастосунку для оцінки психоемоційного стану користувачів із використанням методів штучного інтелекту». У межах дослідження здійснено аналіз предметної області, сформовано вимоги до архітектури системи та обрано технологічний стек, що включає React/TypeScript для клієнтської частини, Node.js/Express для серверної логіки, PostgreSQL/Prisma для зберігання даних та модель Llama 3.3-

70b через Groq API для інтелектуального аналізу. Розроблено математичну модель зваженого скорингу з інверсією балів для тестів благополуччя, реалізовано модуль промпт-інжинірингу з вбудованими етичними обмеженнями, а також механізм резервної генерації рекомендацій у разі недоступності ШІ-сервісу.

Результати виконаної роботи можуть бути використані організаціями для впровадження корпоративних програм підтримки ментального здоров’я, освітніми закладами – для систематичного відстеження психологічного клімату серед учнівської молоді, а також окремими користувачами як інструмент індивідуального самоспостереження та профілактики емоційного виснаження. Окрім того, розроблена архітектура та алгоритмічні підходи можуть слугувати основою для подальших досліджень у галузі застосування штучного інтелекту в превентивній психології, адаптивного тестування та цифрової поведінкової аналітики.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

У цьому розділі здійснено комплексний аналіз предметної області цифрової психологічної допомоги та моніторингу психоемоційного стану. Розглянуто сучасні тенденції розвитку інформаційних систем у сфері ментального здоров’я, проведено критичний огляд існуючих вебплатформ та застосунків для оцінювання стресу й тривожності, виявлено їхні переваги та обмеження. Окрему увагу приділено можливостям застосування методів штучного інтелекту, зокрема обробки природної мови та великих мовних моделей, для автоматизованої семантичної інтерпретації психометричних даних. На основі проведеного дослідження сформульовано ключові прогалини сучасних рішень, обґрунтовано доцільність розроблення власного вебзастосунку та визначено вимоги до його архітектури, функціоналу й безпеки даних. Результатом розділу є чітке формулювання об’єкта, мети та переліку завдань кваліфікаційної роботи, що створює методологічне підґрунтя для подальшого проєктування, програмної реалізації та тестування системи.

1.1. Сучасний стан проблеми оцінки психоемоційного стану та цифрової психологічної допомоги

У сучасних умовах стрімкої цифровізації, глобальних трансформацій ринку праці та підвищення когнітивного навантаження в освітньому середовищі, проблеми збереження ментального здоров’я набувають системного характеру. За офіційними даними Всесвітньої організації охорони здоров’я, рівень хронічного стресу, тривожних розладів та професійного вигорання демонструє стійку тенденцію до зростання, особливо серед молоді та працівників інтенсивних секторів економіки [1]. Тривалий психоемоційний дистрес не лише знижує якість життя, а й негативно впливає на когнітивні функції, продуктивність праці, академічну успішність та соціальну адаптацію.

Виявлення цих станів на ранніх стадіях, організація превентивного моніторингу та забезпечення доступу до профілактичних заходів визнаються пріоритетними напрямами розвитку суспільної охорони здоров’я та корпоративної культури.

Традиційні підходи до оцінки психоемоційного стану ґрунтуються на клінічних інтерв’ю, спостереженні фахівців та стандартизованих паперових опитувальниках, таких як шкала сприйнятого стресу (PSS), опитувальник тривожності Спілбергера-Ханіна, шкала депресії Бека або опитувальник вигорання Маслач [2 - 4].

Ці інструменти мають високу психометричну валідність у контрольованих умовах, проте їх практичне застосування в масштабах організацій чи навчальних закладів супроводжується суттєвими обмеженнями. До них належать: висока трудомісткість обробки даних, відсутність можливості відстеження динаміки стану в реальному часі, залежність результатів від суб’єктивного сприйняття респондента в момент заповнення, а також логістичні та фінансові бар’єри для масового впровадження. Крім того, класичні методи не передбачають автоматизованої інтерпретації відповідей із урахуванням індивідуального контексту, історії попередніх тестувань та комплексної генерації персоналізованих рекомендацій.

Ці фактори зумовили активний перехід до цифрових форматів психологічного моніторингу. За останнє десятиліття сформувалося кілька основних категорій цифрових рішень: платформи телемедицини та онлайн-терапії, мобільні застосунки для самодопомоги (self-help), корпоративні програми підтримки співробітників (Employee Assistance Programs, EAP) та освітні аналітичні системи [5]. Більшість із них інтегрує стандартизовані онлайн-тести, журнали настрою, техніки релаксації або можливість консультацій із фахівцями.

Попри широку доступність, аналіз функціоналу існуючих платформ виявляє ряд суттєвих недоліків. По-перше, переважна кількість застосунків пропонує статичну інтерпретацію результатів на основі жорстких порогових

значень, що не враховує нюанси відповіді, когнітивні спотворення або зміни стану протягом тривалого періоду. По-друге, системи рідко забезпечують зворотний зв’язок у формі структурованих, контекстно-адаптованих рекомендацій, обмежуючись загальними порадами або посиланнями на зовнішні ресурси. По-третє, багато рішень не відповідають сучасним вимогам до конфіденційності та анонімності даних, що є критичним фактором для корпоративного та освітнього секторів, де працівники або учні можуть уникати тестування через страх стигматизації чи негативних наслідків для кар’єри/навчання.

Окремої уваги потребує питання етичної та методологічної безпеки автоматизованих систем у сфері ментального здоров’я. Існують ризики надмірної залежності користувачів від алгоритмічних оцінок, некоректної інтерпретації симптомів, відсутності чітких дисклеймерів щодо немедичного характеру рекомендацій, а також використання даних без належного інформованого consent. Ці аспекти підкреслюють необхідність розроблення платформ, які поєднують науково обґрунтовані психометричні інструменти з прозорими алгоритмами обробки, вбудованими механізмами безпеки та чіткою комунікацією меж застосування системи.

Стрімкий розвиток технологій штучного інтелекту, зокрема великих мовних моделей (LLM) та методів обробки природної мови (NLP), відкриває нові можливості для подолання зазначених обмежень [6 - 8].

Сучасні генеративні моделі здатні аналізувати не лише кількісні бали, а й семантичну структуру відповідей, виявляти приховані патерни емоційного напруження, формувати індивідуалізовані висновки та генерувати структуровані поради з урахуванням категорії тесту, рівня серйозності стану та історії взаємодії [9].

Інтеграція ШІ-модулів у вебзастосунки дозволяє реалізувати динамічну інтерпретацію даних, забезпечити масштабованість системи для тисяч користувачів одночасно та підтримувати багаторівневу архітектуру контролю доступу. Проте ефективне застосування ШІ в психологічній діагностиці

вимагає ретельного налаштування системних інструкцій (prompt engineering), впровадження механізмів fallback-генерації на випадок недоступності сервісів, суворих етичних фільтрів та валідації результатів на контрольних вибірках.

Аналіз сучасного стану цифрової психології свідчить, що попри наявність численних платформ для самооцінювання, на ринку відсутнє цілісне рішення, яке б одночасно забезпечувало: валідовану цифрову адаптацію стандартизованих опитувальників, інтелектуальну інтерпретацію результатів із застосуванням сучасних LLM, персоналізовану генерацію рекомендацій з урахуванням рівня стресу/тривоги/виснаження, багаторівневу систему авторизації, анонімність збору даних та повну відповідність вимогам інформаційної безпеки. Вирішення цієї проблеми потребує створення клієнт-серверного вебзастосунку, архітектура якого передбачає модульну взаємодію між інструментами збору даних, математичним ядром скорингу, ШІ-сервісом аналізу та інтерфейсом візуалізації аналітики. Така система має бути орієнтована на практичне застосування в корпоративному та освітньому середовищі, де масовий скринінг, своєчасне виявлення груп ризику та профілактична підтримка є критично важливими компонентами управління людськими ресурсами.

Таким чином, сучасний стан проблеми оцінки психоемоційного стану характеризується переходом від локальних, суб’єктивних та трудомістких методів до цифрових, автоматизованих та масштабованих рішень. Попри прогрес у розробці онлайн-тестів та застосунків самодопомоги, існують суттєві прогалини в сфері інтелектуальної інтерпретації даних, етично безпечної генерації рекомендацій та інтеграції системи в організаційні процеси моніторингу.

Подолання цих обмежень можливе шляхом поєднання психометрично валідних інструментів з сучасними методами штучного інтелекту, що обумовлює доцільність розроблення власного вебзастосунку з архітектурою, спрямованою на безпеку, персоналізацію та аналітичну гнучкість.

1.2. Огляд існуючих вебзастосунків та платформ для моніторингу стресу

Аналіз сучасного ринку цифрових інструментів для підтримки ментального здоров’я демонструє стрімке зростання кількості вебплатформ та мобільних застосунків, орієнтованих на оцінювання та моніторинг психоемоційного стану. Існуючі рішення можна умовно класифікувати на чотири основні категорії: платформи самодопомоги та медитації, корпоративні програми підтримки співробітників (Employee Assistance Programs, EAP), ШІ-чатботи та діагностичні сервіси, а також академічні прототипи й відкриті інструменти [10]. Кожна з цих категорій має власні переваги, проте жодна не забезпечує повноцінного циклу «валідоване тестування → інтелектуальна інтерпретація → персоналізовані рекомендації → анонімний корпоративний/освітній моніторинг».

Платформи самодопомоги та медитації, такі як Headspace, Calm або Insight Timer, орієнтовані переважно на профілактику стресу через аудіо- та відеоконтент, техніки усвідомленості та щоденні журнали настрою. Їхній функціонал обмежується суб’єктивним самооцінюванням без застосування стандартизованих психометричних шкал.

Відсутність структурованої діагностики унеможливлює формування об’єктивної динаміки стану, а алгоритми рекомендацій базуються на шаблонних сценаріях, що не адаптуються до індивідуального рівня тривожності чи вигорання. Крім того, ці системи працюють за моделлю платної підписки, що ускладнює їх масове впровадження в організаційному середовищі.

Корпоративні EAP-платформи (Culture Amp, Qualtrics Employee Experience, Wellable) зосереджені на агрегації анонімних опитувань та формуванні загальної аналітики для HR-відділів. Вони часто інтегрують стандартизовані питання, проте інтерпретація результатів відбувається на рівні агрегованих показників, без індивідуального зворотного зв’язку для співробітника. Багато таких систем використовують жорсткі порогові значення

без урахування контексту, історії проходження або когнітивних особливостей респондента. Окремою проблемою є недостатня прозорість механізмів захисту даних: працівники часто сумніваються в абсолютній анонімності, що знижує чесність відповідей та валідність отриманих метрик.

ШІ-чатботи та діагностичні помічники (Woebot, Wysa, Youper) використовують елементи обробки природної мови та когнітивно-поведінкової терапії (CBT) для підтримки користувачів у реальному часі. Попри високий рівень залучення, їхні діагностичні можливості обмежуються короткими сесіями та базовим сентимент-аналізом.

Відсутність інтеграції з валідованими психометричними опитувальниками унеможливлює кількісну оцінку стану [11]. Більшість таких рішень розроблені виключно англійською мовою, не адаптовані до українських соціокультурних реалій та не містять механізмів резервної генерації відповідей на випадок недоступності ШІ-сервісу. Крім того, етичні фільтри та дисклеймери про немедичний характер порад часто розміщені неявно або ігнорують користувачами.

Академічні та дослідницькі прототипи, представлені у наукових публікаціях останніх років, часто демонструють високу точність алгоритмів класифікації емоцій або прогнозування стресу. Проте вони рідко містять готову до розгортання клієнт-серверну архітектуру, належну систему контролю доступу, валідацію даних або інструкції з безпеки. Такі рішення залишаються ізольованими експериментами без адаптації до реальних умов експлуатації в компаніях чи навчальних закладах.

Для систематизації виявлених особливостей проведено порівняльний аналіз репрезентативних рішень за ключовими критеріями (табл 1.1).

Таблиця 1.1 – Порівняльний аналіз існуючих платформ для моніторингу стресу

Платформа	Тип рішення	Інтеграція ШІ	Психометрична валідність	Анонімність та безпека даних	Персоналізація рекомендацій	Основні обмеження
1	2	3	4	5	6	7
Headspace / Calm	Самодопомога та медитація	Відсутня або базовий сентимент-аналіз	Не застосовується	Висока (шифрування, GDPR)	Шаблонна, на основі підписки	Відсутність діагностики, платна модель, відсутність корпоративної інтеграції
Culture Amp / Qualtrics	Корпоративна EAP-аналітика	Мінімальна (агрегація даних)	Часткова (стандартизовані опитування)	Середня (залежить від політики компанії)	Відсутня на рівні користувача	Відсутність індивідуального зворотного зв’язку, ризик стигматизації, висока вартість ліцензій

Продовження таблиці 1.1

1	2	3	4	5	6	7
Woebot / Wysa	ШІ-чатбот для самодопомоги	NLP + CBT-діалоги	Не застосовується	Середня (дані зберігаються в хмарі провайдера)	Динамічна, але обмежена контекстом сесії	Відсутність структурованих тестів, англомовний інтерфейс, відсутність fallback-логіки
Дослідницькі прототипи	Наукові/експериментальні системи	Глибоке навчання, класифікація, прогнозування	Залежить від джерела даних	Низька (часто відкриті репозиторії)	Відсутня або локальна	Відсутність production-архітектури, безпеки, масштабованості та інструкції користувача
Розроблювана система	Вебзастосунок для оцінки та ШІ-аналізу	LLM (Llama 3.3-70b) + сентимент-аналіз + скоринг	Повна (адаптовані PSS, STAI, MBI, WELLNESS)	Висока (JWT, Helmet, rate limiting, анонімність)	Динамічна, severity-based, з fallback-модулем	Інтегрує тестування, ШІ-інтерпретацію, багаторівневий доступ та етичні фільтри

Аналіз існуючих рішень дозволяє виокремити низку критичних прогалин, які обмежують їх практичне застосування в організаційному та освітньому середовищі:

− відсутність цілісного циклу оцінювання. Більшість платформ розділяють функції тестування та рекомендацій. Користувач проходить опитувальник, але отримує лише статичний бал без контекстної інтерпретації або адаптованих порад;

− недостатня гнучкість ШІ-інтерпретації. Існуючі системи використовують жорсткі правила або шаблонні відповіді, що не враховують індивідуальний контекст, історію змін стану або когнітивні особливості респондента. Відсутні механізми резервної генерації (fallback) на випадок недоступності ШІ-провайдера;

− проблеми конфіденційності та анонімності. Корпоративні рішення часто вимагають реєстрації з прив’язкою до робочого email, що викликає недовіру співробітників. Багато сервісів не реалізують багаторівневу модель доступу з чітким розділенням даних користувачів та адміністраторів;

− відсутність локалізації та психометричної адаптації. Переважна кількість рішень орієнтована на англомовну аудиторію та не враховує культурно-соціальні особливості українських користувачів, а також нормативні вимоги до психологічного моніторингу в Україні;

− обмежені етичні гарантії. Системи рідко містять вбудовані фільтри, що запобігають формуванню псевдомедичних висновків, не містять чітких дисклеймерів про інформаційний характер порад або автоматично не пропонують контакти кризової підтримки при високих показниках стресу.

Подолання зазначених обмежень вимагає створення цілісної клієнт-серверної вебсистеми, яка поєднує валідовані цифрові опитувальники, модуль інтелектуальної інтерпретації на базі сучасних LLM, багаторівневу авторизацію, механізми анонімного збору даних, суворі етичні фільтри та резервну логіку генерації рекомендацій. Саме такий підхід реалізовано в розроблюваному вебзастосунку, архітектура якого спрямована на забезпечення

масштабованості, конфіденційності та адаптивності рекомендаційної системи для корпоративного та освітнього середовища.

1.3. Можливості застосування методів штучного інтелекту в аналізі тестових та текстових відповідей

Сучасний розвиток галузі штучного інтелекту, зокрема методів обробки природної мови (NLP) та великих мовних моделей (LLM), відкриває нові можливості для автоматизованого аналізу психологічних даних [12]. Традиційні алгоритмічні підходи до інтерпретації результатів тестування обмежуються жорсткими математичними формулами та статичними пороговими значеннями, що не дозволяє враховувати контекст, індивідуальні особливості відповідей чи динаміку змін стану респондента протягом часу. Інтеграція генеративних моделей у процес обробки відповідей дозволяє перейти від простого підрахунку балів до семантичного аналізу та формування структурованих, персоналізованих висновків, адаптованих до конкретного рівня емоційного навантаження.

Аналіз структурованих тестових відповідей (наприклад, за шкалами Лікерта) із застосуванням ШІ здійснюється не лише через математичне зважування балів, а й через виявлення патернів у комбінаціях відповідей. Сучасні LLM здатні зіставляти кількісні показники з якісними характеристиками, формувати багатовимірну оцінку стану (одночасно оцінюючи рівень стресу, тривожності та емоційного виснаження) та генерувати інтерпретації, що враховують категорію тесту [13]. У розроблюваній системі реалізовано алгоритм інверсії балів для тестів благополуччя (WELLNESS), де низькі кількісні показники автоматично трансформуються у високі рівні ризику, а ШІ-модуль враховує цю логіку під

час формування підсумкового звіту, що забезпечує коректну семантичну відповідність між математичним скорингом та текстовим описом стану [14].

Обробка відкритих або контекстно збагачених відповідей потребує застосування методів промпт-інжинірингу. Системні інструкції для моделей налаштовуються таким чином, щоб забезпечити підтримувальний, нестигматизувальний тон, уникати клінічних термінів (наприклад, «депресія», «розлад», «патологія») та чітко позначати інформаційний характер аналізу. У практичній реалізації використано структуровані системні промпти, що містять правила безпеки, вимоги до формату виведення (JSON) та обов'язкові елементи (дисклеймери, рекомендації щодо звернення до фахівця). Це дозволяє стандартизувати вивід генеративної моделі, мінімізувати ризик генерації некоректних порад та забезпечити відповідність результатів етичним стандартам цифрової психологічної підтримки.

Важливим аспектом є технічна реалізація інтеграції ШІ у вебзастосунок. Використовується архітектура з абстрактним інтерфейсом ШІ-провайдера, що забезпечує гнучкість заміни моделей (наприклад, Llama 3.3-70b через Groq API, OpenAI або DeepSeek) без зміни бізнес-логіки системи [15, 16]. Оскільки зовнішні ШІ-сервіси можуть бути тимчасово недоступними або перевищувати ліміти запитів, реалізовано механізм резервної генерації (fallback) [17]. У разі помилки виклику зовнішнього API система автоматично перемикається на локальний алгоритм, який формує рекомендації на основі розрахованих процентів та категоризації серйозності (LOW, MILD, MODERATE, HIGH, SEVERE). Це забезпечує безперервність роботи системи, стабільність користувацького досвіду та дотримання принципу відмовостійкості в критичних сценаріях взаємодії.

Окремої уваги заслуговують аспекти етичної безпеки та валідації результатів. Усі згенеровані звіти містять обов'язкове застереження про немедичний характер інформації та рекомендацію звернутися до кваліфікованого психолога при високих показниках дистресу. Додатково інтегрується оцінка впевненості моделі (confidence score), яка дозволяє

користувачу орієнтуватися в ступені надійності ШІ-аналізу [18]. При критичних значеннях автоматично додаються контакти кризових гарячих ліній, що відповідає сучасним стандартам цифрової психологічної допомоги та запобігає ризику залишення користувача без підтримки у гострих станах.

Попри широкі можливості, застосування ШІ в психологічній діагностиці має об’єктивні обмеження. Генеративні моделі схильні до артефактної генерації «галюцинацій», можуть відтворювати культурні або мовні упередження, присутні у навчальних даних, та не здатні повноцінно замінити клінічне інтерв’ю або професійну терапію [19]. Тому ефективність системи забезпечується комбінацією валідованих психометричних методик, детально налаштованих промптів з етичними фільтрами, механізму fallback та чіткої комунікації меж застосування інструменту. ШІ виступає не як діагност, а як інтелектуальний помічник для попереднього скринінгу, інтерпретації тенденцій та формування структурованих рекомендацій з самодопомоги.

Таким чином, сучасні методи штучного інтелекту дозволяють автоматизувати семантичний аналіз тестових даних, генерувати адаптовані рекомендації та інтегрувати етично безпечні механізми підтримки у веб-середовище. Однак для реалізації цих можливостей у корпоративному або освітньому середовищі необхідна цілісна клієнт-серверна архітектура з багаторівневим контролем доступу, валідацією вхідних даних, захищеним зберіганням результатів та модульною структурою для масштабування. Це зумовлює необхідність постановки конкретної задачі щодо розроблення вебзастосунку, що об’єднує зазначені технологічні та методологічні рішення в єдину функціональну систему.

1.4. Постановка задачі: актуальність, об’єкт, мета та перелік завдань роботи

Стрімке зростання рівня хронічного стресу, тривожності та професійного вигорання в сучасному суспільстві, особливо в корпоративному та освітньому середовищах, обумовлює гостру потребу у масштабованих інструментах превентивного моніторингу психоемоційного стану. Існуючі цифрові рішення часто обмежуються статичною агрегацією тестів без глибокої інтерпретації результатів, не забезпечують персоналізованого зворотного зв’язку або ігнорують вимоги конфіденційності, анонімності та етичної безпеки даних. Інтеграція сучасних методів штучного інтелекту, зокрема великих мовних моделей та алгоритмів сентимент-аналізу, дозволяє подолати ці обмеження, автоматизувати семантичну інтерпретацію відповідей, адаптувати рекомендації до індивідуального рівня навантаження та забезпечити безперервність сервісу за рахунок механізмів резервної генерації [20]. Це обумовлює доцільність створення цілісної клієнт-серверної системи, орієнтованої на практичне застосування в організаціях та навчальних закладах.

Об’єктом роботи є процес розроблення клієнт-серверного вебзастосунку для діагностики та моніторингу психоемоційного стану користувачів.

Метою роботи є розроблення вебзастосунку, що автоматизує проходження валідованих психометричних тестів, аналізує результати за допомогою великої мовної моделі Llama 3.3-70b та генерує персоналізовані рекомендації для покращення психологічного благополуччя.

Для досягнення мети необхідно вирішити такі завдання:

– проаналізувати сучасні підходи до цифрової психологічної допомоги та виявити обмеження існуючих платформ для моніторингу стресу;

– сформулювати функціональні та архітектурні вимоги до системи, обрати технологічний стек та психометричні методики для оцінювання стресу, тривожності, вигорання та благополуччя;

– розробити математичну модель зваженого скорингу з інверсією балів для тестів благополуччя та алгоритм інтелектуальної інтерпретації результатів із використанням промпт-інжинірингу й вбудованих етичних фільтрів;

– реалізувати клієнт-серверну архітектуру з модулями аутентифікації, тестування, інтерактивної аналітики, багаторівневим контролем доступу та механізмом резервної генерації рекомендацій;

– інтегрувати модуль штучного інтелекту на базі Llama 3.3-70b через Groq API, забезпечивши валідацію вхідних даних, конфіденційність та відповідність стандартам інформаційної безпеки;

– провести функціональне, юзабіліті- та навантажувальне тестування вебзастосунку, оцінити точність ШІ-аналізу на контрольній вибірці та сформулювати перспективи подальшого розвитку системи.

2 МАТЕМАТИЧНЕ ОБҐРУНТУВАННЯ ВИБРАНИХ МЕТОДІВ ТА ОСОБЛИВОСТІ РОЗРОБЛЕННЯ МОДЕЛІ АНАЛІЗУ

У цьому розділі здійснено математичне та алгоритмічне обґрунтування методів, покладених в основу розроблення модуля інтелектуального аналізу психоемоційного стану. Розглянуто математичні моделі первинного скорингу, нормалізації та інверсії показників для тестів різної семантичної спрямованості, що забезпечує коректну агрегацію результатів незалежно від типу опитувальника. Деталізовано алгоритми обробки структурованих відповідей, принципи промпт-інжинірингу для великих мовних моделей, механізми резервної генерації порад та валідації вхідних даних. Наведено об’єктно-орієнтоване моделювання взаємодії компонентів системи, яке формалізує логіку клієнт-серверного обміну, багаторівневої авторизації та інтеграції ШІ-провайдерів, створюючи методологічне підґрунтя для програмної реалізації, тестування та оцінювання ефективності системи в третьому розділі.

2.1. Психометричні основи та валідні методики вимірювання рівня стресу/емоційного стану

Ефективність будь-якої автоматизованої системи оцінювання психоемоційного стану безпосередньо залежить від якості та наукової обґрунтованості інструментів збору даних. У контексті розроблення вебзастосунку критично важливим є поєднання класичних психометричних підходів з цифровими алгоритмами обробки. Психометрика як наукова дисципліна базується на трьох фундаментальних характеристиках вимірювальних інструментів: валідності, надійності та стандартизації. Валідність визначає, наскільки точно тест вимірює саме ту конструкцію, для

якої він призначений (наприклад, рівень сприйнятого стресу або емоційного вигорання). Надійність характеризує узгодженість результатів при повторному тестуванні або внутрішню консистентність пунктів опитувальника. Стандартизація забезпечує єдині умови проведення, інтерпретації та порівняння результатів між різними групами респондентів.

Для інтеграції у розроблювану систему було обрано та адаптовано чотири валідовані методики, які охоплюють ключові аспекти емоційного стану: шкалу сприйнятого стресу (PSS), опитувальник ситуативної тривожності за Спілбергером-Ханіним (STAI-S), опитувальник емоційного вигорання (MBI) та шкалу психологічного благополуччя (WELLNESS). Кожна з методик використовує ординальну шкалу Лікерта, що дозволяє квантувати суб'єктивні відчуття у числовому форматі. У цифровому середовищі це реалізовано через інтерактивні компоненти інтерфейсу, які фіксують вибір користувача, обмежують час проходження (за необхідності) та запобігають пропуску запитань, що мінімізує ризик випадкових або неповних відповідей.

Математична модель первинного скорингу базується на адитивному накопиченні зважених відповідей. Нехай тест складається з nзапитань, кожне з яких має kваріантів відповідей, пронумерованих від 0 до $k-1$. Відповідь користувача на i-те запитання позначається як $v_i \in \{0,1,\dots,k-1\}$. Вага w_iвідображає ступінь впливу конкретного пункту на загальний конструкт (за замовчуванням $w_i = 1$для всіх пунктів у стандартних коротких версіях тестів). Загальний сирий бал S обчислюється за формулою:

$$S = \sum_{i=1}^{n} v_i \cdot w_i. \qquad (2.1)$$

Максимально можливий бал $S_{\max}$ визначається як сума добутків максимальних значень варіантів відповідей на їхні ваги:

$$S_{\max} = \sum_{i=1}^{n} (k_i - 1) \cdot w_i. \quad (2.2)$$

Для уніфікації інтерпретації між тестами з різною кількістю запитань та шкалами здійснюється нормалізація результату у відсотковому діапазоні [0;100]. Нормалізований показник Pрозраховується відповідно до виразу:

$$P = \frac{S}{S_{\max}} \cdot 100\%. \quad (2.3)$$

Особливістю математичної моделі є врахування семантичної спрямованості тестів. Для методик, що вимірюють негативні стани (стрес, тривога, вигорання), високе значення P прямо корелює з підвищеним рівнем дистресу. Однак для тесту благополуччя (WELLNESS) логіка оцінювання є оберненою: низькі бали свідчать про незадоволеність, відсутність енергії або підтримки, тоді як високі вказують на стабільний стан. Для коректної інтеграції цього тесту в єдину шкалу серйозності застосовується операція інверсії:

$$P_{\text{inv}} = 100\% - P. \quad (2.4)$$

Отриманий нормалізований показник P^* (де $P^* = P$ для тестів дистресу та $P^* = P_{\text{inv}}$ для тесту благополуччя) використовується як вхідний параметр для функції визначення рівня серйозності $f_{\text{severity}}(P^*)$, яка реалізує кусково-постійну логіку категоризації:

$$f_{\text{severity}}(P^*) = \begin{cases} \text{LOW}, & 0\% \le P^* < 20\% \\ \text{MILD}, & 20\% \le P^* < 40\% \\ \text{MODERATE}, & 40\% \le P^* < 60\% \\ \text{HIGH}, & 60\% \le P^* < 80\% \\ \text{SEVERE}, & 80\% \le P^* \le 100\% \end{cases}. \quad (2.5)$$

Ця функція транслює безперервну числову шкалу у дискретний набір категорій, що спрощує візуалізацію для користувача та слугує структурованим контекстом для подальшого аналізу модулем штучного інтелекту. Важливо зазначити, що математичний скоринг не є фінальним діагностичним висновком. Він виступає лише етапом первинної обробки, який забезпечує консистентність даних, фільтрує статистичні артефакти та формує уніфікований вектор ознак для семантичної інтерпретації.

Цифрова адаптація обраних методик також враховує специфіку взаємодії користувача з інтерфейсом. Для запобігання ефекту «центрування» (схильності обирати середні варіанти) та «ефекту ореолу» (впливу попередніх відповідей на наступні) реалізовано послідовне відображення запитань з динамічним прогрес-баром, випадковим чергуванням варіантів відповідей (крім шкал Лікерта з чіткою градацією) та обмеженням часу на проходження тестів з категорією складності HARD. Це забезпечує збір більш достовірних даних, які відповідають вимогам до надійності вимірювань у неконтрольованих умовах.

Отримані структуровані показники S, P^*, $f_{\text{severity}}(P^*)$ та історія відповідей $\{v_1, v_2, \dots, v_n\}$ передаються на вхід модулю інтелектуального аналізу. Саме тут відбувається перехід від кількісної психометричної оцінки до якісної семантичної інтерпретації, що вимагає застосування алгоритмів обробки природної мови, промпт-інжинірингу та механізмів резервного генерування порад, які детально розглядаються у наступних підрозділах.

2.2. Методи обробки та аналізу відповідей користувачів

Після здійснення первинного математичного скорингу та нормалізації результатів, отримані числові показники передаються до модуля інтелектуального аналізу. Класичні алгоритмічні підходи обмежуються

статичними пороговими значеннями та не здатні враховувати контекст, комбінації відповідей або індивідуальні патерни поведінки респондента. Для подолання цих обмежень у розроблюваній системі застосовано методи обробки природної мови (NLP) та великі мовні моделі (LLM), що дозволяють здійснити глибокий семантичний аналіз та сформувати структуровані, персоналізовані висновки.

Основним інструментом інтелектуальної обробки є промпт-інжиніринг – технологія формування системних інструкцій для великих мовних моделей [21]. У даній системі реалізовано багаторівневу структуру промпта, що включає ролеве призначення моделі, контекст тесту, вхідні дані та жорсткі правила безпеки. Системний промпт формується динамічно залежно від категорії тесту (STRESS, ANXIETY, BURNOUT, WELLNESS) та містить дев’ять обов’язкових етичних обмежень: заборону на надання медичних діагнозів, використання клінічної термінології (наприклад, «депресія», «розлад», «патологія»), обов’язкове зазначення інформаційного характеру аналізу, рекомендацію звернення до фахівця при високих показниках, підтримувальний нестигматизувальний тон та вимогу відповідати українською мовою. Такий підхід гарантує, що згенеровані висновки відповідають стандартам цифрової психологічної підтримки та не порушують принципів академічної та медичної етики.

Для забезпечення сумісності отриманих від ШІ результатів із архітектурою системи, модель налаштована на генерацію структурованих даних у форматі JSON. Вихідний об’єкт містить наступні поля: stressLevel, anxietyLevel, burnoutLevel (числові значення у діапазоні 0–100), emotionalState (короткий опис стану), summary (підтримувальний підсумок), recommendations[10] (масив персоналізованих порад з категорією та пріоритетом) та confidence (рівень впевненості моделі у діапазоні 0–1). Оскільки генеративні моделі можуть повертати відповідь із додатковими розмітковими тегами (наприклад, json), реалізовано механізм очищення та парсингу: виконується пошук першої фігурної дужки та останньої, виділення

підстроки та спроба десеріалізації за допомогою стандартного методу розбору JSON. У разі синтаксичної помилки або відсутності валідної структури активовується резервний сценарій обробки.

Механізм резервної генерації (fallback) є критичним компонентом відмовостійкості системи. Якщо зовнішній ШІ-провайдер недоступний, перевищено ліміт запитів або результат парсингу виявився некоректним, система автоматично перемикається на локальний алгоритм формування рекомендацій. Цей алгоритм базується на раніше розрахованому нормалізованому показнику P^* та функції категоризації серйозності $f_{\text{severity}}(P^*)$. Для кожного рівня (LOW, MILD, MODERATE, HIGH, SEVERE) заздалегідь визначено шаблони емоційних станів, підсумків та набори рекомендацій, згрупованих за категоріями (relaxation, lifestyle, mindfulness, social, professional, crisis, wellness). Пріоритет порад динамічно коригується залежно від рівня серйозності: від підтримки здорових звичок при низьких показниках до наполегливої рекомендації звернення до психолога та контактів кризових гарячих ліній при критичних значеннях. Такий підхід гарантує безперервність сервісу незалежно від стану зовнішніх API-сервісів.

Інтеграція ШІ-модуля реалізована за допомогою патерну абстрактного провайдера, що забезпечує гнучкість та масштабованість системи. Базовий інтерфейс провайдера визначає уніфікований метод прийому тексту промпта та системних інструкцій, а повертає текст відповіді та метрики використання обчислювальних токенів. Реалізовано три конкретні провайдери, що взаємодіють з сервісами OpenAI, DeepSeek та Groq (основний, з моделлю Llama 3.3-70b). Вибір провайдера здійснюється на етапі ініціалізації сервісу на основі конфігураційних змінних оточення, що дозволяє без зміни бізнес-логіки перемикатися між сервісами або підключати нові. Усі запити супроводжуються логуванням, обробкою помилок мережі та контролюванням швидкості викликів через проміжне програмне забезпечення обмеження частоти запитів на рівні сервера.

Важливим аспектом аналізу є семантичне узгодження математичного скорингу та ШІ-інтерпретації. Оскільки модель аналізує лише текстовий масив відповідей та інструкцію щодо інтерпретації, існує ризик розбіжності між розрахованим відсотком P та згенерованими рівнями стресу чи тривоги. Для усунення цього недоліку реалізовано процедуру злиття результатів: якщо ШІ-модуль повертає некоректні або відсутні числові показники, вони автоматично замінюються значеннями, розрахованими локальним генератором на основі P^*. Підсумковий звіт завжди доповнюється дисклеймером про немедичний характер аналізу. Рівень впевненості за замовчуванням встановлюється на 0.6 для локального генератора та 0.8–1.0 для успішної ШІ-генерації, що дозволяє візуалізувати ступінь надійності висновків у користувацькому інтерфейсі.

Таким чином, комбінація промпт-інжинірингу з жорсткими етичними фільтрами, структурованого парсингу даних, патерну абстрактного провайдера та надійного механізму резервної генерації формує цілісну логіку обробки та аналізу відповідей. Отримані структуровані дані передаються до модуля формування індивідуальних рекомендацій, алгоритми категоризації, пріоритизації та збереження яких детально описуються у наступному підрозділі.

2.3. Математичні моделі та алгоритми формування індивідуальних рекомендацій

Формування індивідуальних рекомендацій є завершальним етапом інтелектуальної обробки результатів психоемоційного тестування. На відміну від статичних порад, що базуються виключно на сумарному балі, запропонована система використовує гібридну модель, яка поєднує детермінований алгоритмічний підхід із можливостями генеративного штучного інтелекту. Це забезпечує високу надійність сервісу, масштабованість

інтерпретації та відповідність сучасним вимогам до відмовостійкості інформаційних систем.

Основою математичної моделі генерації рекомендацій є функція відображення нормалізованого показника P^* у множину категоризованих порад $\mathcal{R}$. Нехай $\mathcal{C} = \{c_1, c_2, \dots, c_k\}$– скінченна множина категорій рекомендацій, де кожен елемент відповідає певному напрямку підтримки (наприклад, relaxation, lifestyle, mindfulness, social, professional, crisis, wellness). Кожна рекомендація $r \in \mathcal{R}$ визначається кортежем:

$$r = (\text{title,content,category,priority}), \text{priority} \in [1,5]. \quad (2.6)$$

Алгоритм формування вихідного набору $\mathcal{R}_{\text{out}}$ базується на кусково-постійній функції серйозності $f_{\text{severity}}(P^*)$, визначеній у підрозділі 2.1. Для кожного рівня $s \in \{\text{LOW,MILD,MODERATE,HIGH,SEVERE}\}$ існує попередньо визначена база шаблонів $\mathcal{B}_s \subset \mathcal{R}$. Вибір конкретного підмножини рекомендацій здійснюється за правилом:

$$\mathcal{R}_{\text{det}} = \mathcal{B}_{f_{\text{severity}}(P^*)} \cup \mathcal{B}_{\text{base}}, \quad (2.7)$$

де $\mathcal{B}_{\text{base}}$ – базовий набір універсальних порад, який включається незалежно від рівня серйозності для підтримки профілактичного характеру платформи.

Пріоритет кожної рекомендації визначається функцією $\phi(s, c)$, яка залежить від рівня серйозності sта категорії c. Для категорій, пов'язаних із критичною підтримкою, встановлюється обов'язковий максимальний пріоритет $p_{\max} = 5$ при виявленні ознак високого дистресу. Математично це представлено у вигляді:

$$p(r) = \begin{cases} 5, & \text{якщо } c(r) = \text{'crisis'} \wedge s \in \{\text{HIGH,SEVERE}\} \\ \pi(s, c(r)), & \text{в іншому випадку} \end{cases}, \quad (2.8)$$

де π– матриця вагових коефіцієнтів, визначена на основі психологічних рекомендацій щодо превентивної підтримки та структурована таким чином, щоб забезпечити поступове зростання важливості категорій professional та lifestyle зі збільшенням s.

Коли ШІ-модуль успішно повертає валідну відповідь у форматі JSON, його рекомендації $\mathcal{R}_{\text{AI}}$ об'єднуються з детермінованим набором $\mathcal{R}_{\text{det}}$. Для уникнення дублювання та забезпечення семантичної релевантності використовується функція злиття $\mathcal{M}(\mathcal{R}_{\text{AI}}, \mathcal{R}_{\text{det}})$, яка пріоритезує ШІ-генерацію за наявності повного набору обов'язкових полів, але завжди включає критичні категорії з локальної бази. Рівень впевненості системи γ розраховується як:

$$\gamma = \begin{cases} \gamma_{\text{AI}}, & \text{якщо AI-відповідь валідна та відповідає схемі валідації} \\ 0.6, & \text{якщо активовано fallback-режим} \end{cases}, \quad (2.9)$$

де $\gamma_{\text{AI}} \in [0.8, 1.0]$ – оцінка впевненості, що генерується моделлю або встановлюється за замовчуванням на етапі пост-процесингу. Цей показник передається на клієнтську сторону для візуалізації ступеня надійності аналітики.

У разі неможливості отримання відповіді від зовнішнього API (мережева помилка, перевищення ліміту запитів, невалідний синтаксис JSON або порушення етичних обмежень у виводі), система автоматично перемикається на локальний алгоритм резервної генерації. Його логіка формалізується у вигляді послідовності кроків:

Етап 1. Отримання нормалізованого показника P^* з модуля математичного скорингу.

Етап 2. Обчислення рівня серйозності $s = f_{\text{severity}}(P^*)$.

Етап 3. Вибір відповідного підмасиву $\mathcal{B}_s$ з локальної бази та об'єднання його з $\mathcal{B}_{\text{base}}$.

Етап 4. Динамічне коригування пріоритетів кожного елемента відповідно до функції $\phi(s, c)$.

Етап 5. Формування підсумку summary шляхом підстановки шаблонного тексту емоційного стану, ін’єкція обов’язкового дисклеймера про немедичний характер аналізу та встановлення $\gamma = 0.6$.

Етап 6. Повернення структурованого об’єкта, сумісного з типом AIAnalysisResult, для подальшого збереження в базі даних та передачі клієнтському інтерфейсу.

Такий підхід гарантує відмовостійкість системи та відповідність принципам безперервності надання інформаційної підтримки навіть за умов тимчасової недоступності зовнішніх ШІ-сервісів.

Окремої уваги заслуговують вбудовані етичні фільтри, які інтегровані в алгоритм на рівні пост-процесингу. При $s \in \{\text{HIGH,SEVERE}\}$ система примусово додає рекомендації з категорій crisis та professional, ігноруючи будь-які спроби ШІ згенерувати поради щодо самодіагностики, зміни медикаментозного лікування або клінічних процедур. Це реалізується через фільтр Ψ, який сканує згенерований набір $\mathcal{R}_{\text{AI}}$ на наявність заборонених патернів (клінічна термінологія, діагностичні формулювання, рекомендації щодо самолікування) та замінює їх на верифіковані шаблони з локальної бази. Формально це можна описати як операцію обмеженої множини:

$$\mathcal{R}_{\text{final}} = \Psi(\mathcal{R}_{\text{AI}}) \cup \mathcal{R}_{\text{crisis}}(s), \text{де } \mathcal{R}_{\text{crisis}}(s) = \emptyset \text{ при } \notin \{\text{HIGH,SEVERE}\}. \quad (2.10)$$

Ця логіка забезпечує дотримання принципів цифрової етики, запобігає ризику стигматизації та гарантує, що користувач завжди отримує безпечну, перевірену та відповідальну інформацію.

Таким чином, поєднання детермінованої математичної моделі категоризації, алгоритму пріоритезації, механізму злиття ШІ-та локальних даних, а також вбудованих етичних обмежень формує надійну основу для генерації персоналізованих рекомендацій. Наступний підрозділ присвячено проєктуванню архітектури вебсистеми, яка інтегрує описані алгоритми в

цілісний клієнт-серверний процес із забезпеченням масштабованості, безпеки даних та модульної взаємодії компонентів.

2.4. Проєктування архітектури вебсистеми та логіки взаємодії модулів

У цьому підрозділі здійснено детальне проєктування архітектурних рішень вебзастосунку для оцінки психоемоційного стану. Розглянуто клієнт-серверну багаторівневу модель, компонентну структуру інтерфейсу, шарову організацію серверної логіки, схему взаємодії модулів та інфраструктуру контейнеризованого розгортання. Особливу увагу приділено забезпеченню модульності, відмовостійкості та інформаційної безпеки на всіх рівнях системи.

Система реалізована за принципом клієнт-серверної архітектури з чітким поділом на три логічні рівні: рівень подання, рівень бізнес-логіки та рівень зберігання даних. Взаємодія між рівнями здійснюється через REST API за протоколом HTTPS із використанням формату JSON для обміну даними. Фронтенд реалізовано як односторінковий застосунок, що забезпечує динамічне оновлення інтерфейсу без повного перезавантаження сторінки. Бекенд виконує функції аутентифікації, валідації запитів, оркестрації бізнес-процесів, інтеграції зі зовнішніми ШІ-сервісами та безпечного доступу до реляційної бази даних. Така модель забезпечує незалежне розгортання компонентів, спрощує тестування окремих модулів та дозволяє масштабувати серверну частину без впливу на клієнтський інтерфейс.

Загальна архітектурна схема системи наведена на рисунку 2.1.

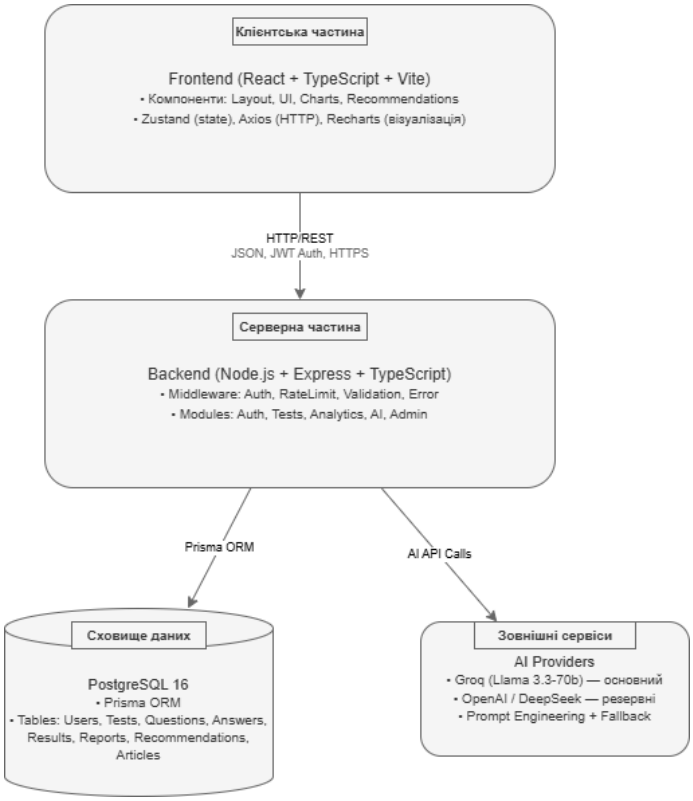

Рисунок 2.1 – Загальна архітектурна схема системи

Інтерфейс побудовано на базі бібліотеки React із використанням мови TypeScript для статичної типізації та зменшення кількості помилок під час розробки. Збірка та оптимізація реалізовано за допомогою інструменту Vite, що забезпечує швидкий гарячий перезапуск та нативну підтримку ES-модулів. Компонентна структура організована ієрархічно: базові універсальні елементи інтерфейсу винесено в окремий каталог, що дозволяє уніфікувати дизайн та зменшити дублювання коду. Маршрутизація реалізована за допомогою React Router, який підтримує захищені маршрути для авторизованих користувачів та адміністраторів, автоматично перенаправляючи неавторизованих осіб на сторінку входу з перевіркою ролей. Управління станом клієнтського застосунку здійснюється за допомогою бібліотеки Zustand. Це рішення обрано через мінімалістичний підхід до зберігання стану, відсутність необхідності у контекстних обгортках та вбудовану підтримку персистентності у локальному сховищі браузера для збереження токенів аутентифікації. Візуалізація аналітичних даних реалізована через бібліотеку Recharts, яка забезпечує

побудову інтерактивних лінійних та стовпчикових діаграм для відображення динаміки рівня стресу, тривожності та вигорання.

Компонентна модель інтерфейсу зображена на рисунку 2.2.

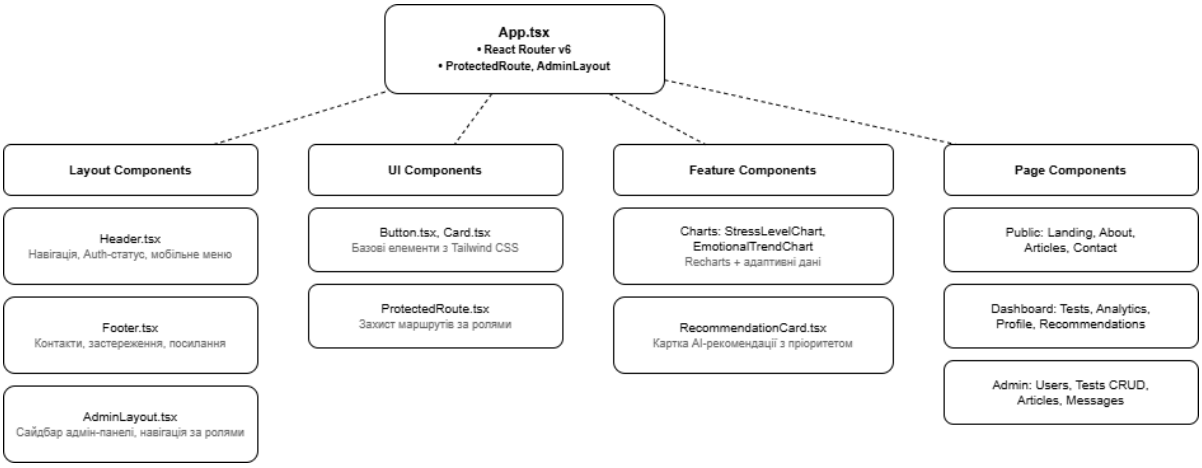

Рисунок 2.2 – Компонентна модель інтерфейсу

Серверна логіка реалізована на базі фреймворку Express у середовищі Node.js із дотриманням принципів шарової архітектури. Запити проходять послідовно через шари: маршрутизатори → контролери → сервіси → шар доступу до даних або зовнішні API. Контролери відповідають лише за обробку HTTP-запитів, валідацію вхідних параметрів через схеми Zod та формування стандартизованих відповідей. Сервіси містять основну бізнес-логіку: розрахунок балів тестів, управління користувачами, генерацію аналітики та взаємодію з ШІ-модулем. Такий поділ спрощує модульне тестування, повторне використання коду та подальшу підтримку системи. Обробка запитів забезпечується ланцюжком проміжного програмного забезпечення: Helmet для встановлення безпечних HTTP-заголовків, CORS для контролю походження запитів, стиснення відповідей, логування через Morgan, обмеження частоти запитів для запобігання атакам перевантаження, а також централізований обробник помилок. Аутентифікація реалізована за моделлю JSON Web Token із короткочасним access-токеном та довгостроковим refresh-токеном, що забезпечує баланс між зручністю та безпекою.

Схема шарової архітектури бекенду представлена на рисунку 2.3.

Як сховище структурованих даних обрано реляційну систему управління базами даних PostgreSQL версії 16, що гарантує ACID-сумісність, підтримку транзакцій, розширені типи даних JSON та високу продуктивність при читанні й записі. Для взаємодії з базою використано ORM-інструмент Prisma, який забезпечує типобезпечну генерацію клієнтських інтерфейсів на основі декларативної схеми, автоматичну міграцію структури бази даних та захист від SQL-ін'єкцій через параметризовані запити. Схема даних включає сутності: користувачі з ролями, опитувальники з категоріями, питання з варіантами відповідей у форматі JSON, відповіді користувача, результати тестування з розрахованими балами та рівнем серйозності, ШІ-звіти, пов'язані з результатом один-до-одного, персоналізовані поради та публікації. Зв'язки між таблицями організовано з каскадним видаленням для забезпечення цілісності даних при видаленні облікових записів або тестів.

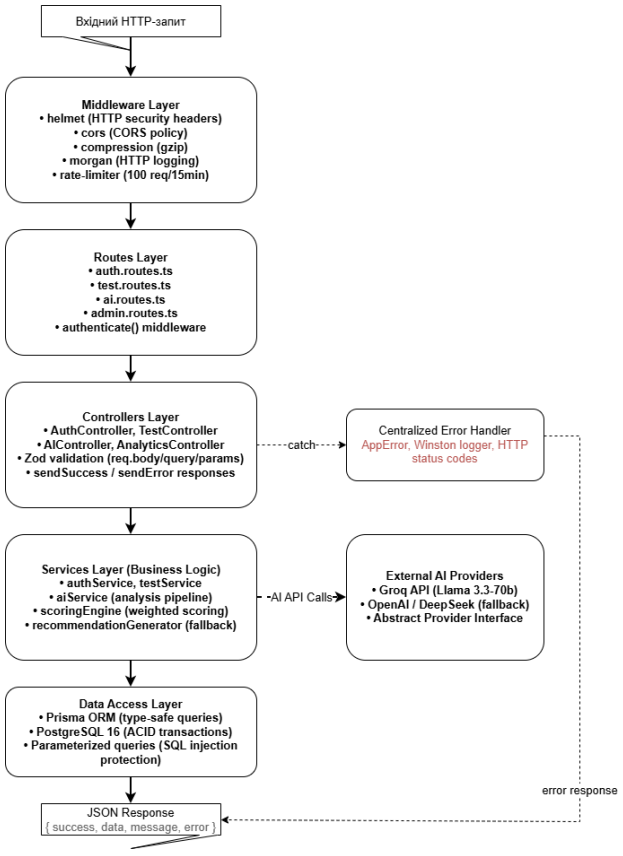

Рисунок 2.3 – Схема шарової архітектури бекенду

ER-діаграма бази даних наведена на рисунку 2.4.

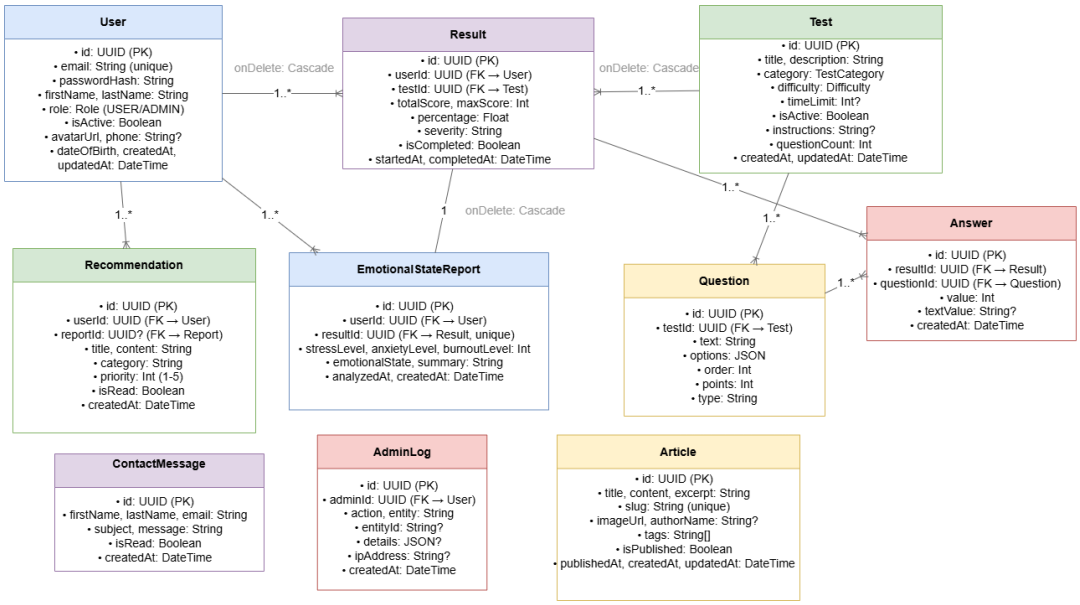

Рисунок 2.4 – ER-діаграма бази даних

Логіка взаємодії модулів під час проходження тестування та отримання аналізу реалізовано за чіткою послідовністю потоків даних. Користувач обирає тест на клієнтській стороні, після чого фронтенд надсилає запит на створення сесії тестування. Сервер повертає ідентифікатор результату та список питань. У процесі проходження кожна відповідь відправляється окремим запитом, що мінімізує ризик втрати даних при тимчасовому розриві мережевого з'єднання.

Після завершення тесту викликається ендпоінт завершення, де сервер обчислює сумарний бал, максимальний можливий бал, процентне співвідношення та, залежно від категорії тесту, застосовує інверсію балів для тесту благополуччя. Результат зберігається в базі даних, після чого фронтенд автоматично викликає ендпоінт ШІ-аналізу. Бекенд формує контекстний промпт на основі відповідей, категорії тесту та розрахованих метрик, надсилає запит до обраного ШІ-провайдера, парсить отриману відповідь у форматі JSON, зливає результати з локальними метриками, зберігає звіт та рекомендації, після чого повертає структуровану відповідь клієнту. У разі помилки мережі або некоректного виводу моделі активовується механізм

резервної генерації, який формує аналогічну відповідь на основі локальних шаблонів та розрахованого рівня серйозності. Така послідовність забезпечує прозорість, відмовостійкість та чіткий контроль за станом даних на кожному етапі.

Діаграма послідовності проходження тесту та ШІ-аналізу зображена на рисунку 2.5.

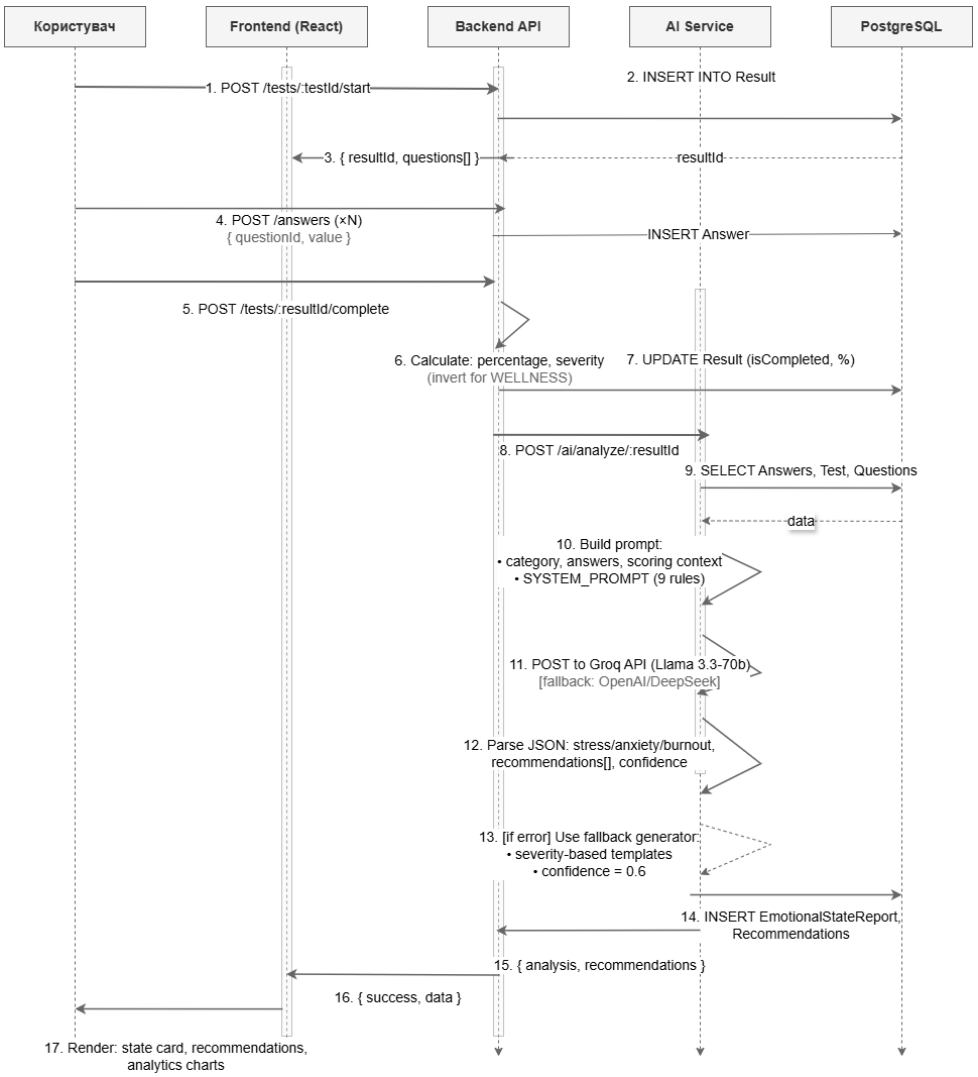

Рисунок 2.5 – Діаграма послідовності проходження тесту та ШІ-аналізу

Для забезпечення відтворюваності середовища та спрощення розгортання використано технологію контейнеризації Docker. Система описана у файлі оркестрації контейнерів, що включає три сервіси: база даних із перевіркою стану, серверна частина з налаштуванням змінних оточення та

логуванням, клієнтська частина на базі Nginx для обслуговування статичних файлів та проксування запитів до API. Така конфігурація ізолює залежності, уніфікує налаштування для локальної розробки та продакшну, а також спрощує горизонтальне масштабування. Безпека системи реалізована на кількох рівнях: валідація всіх вхідних даних, хешування паролів алгоритмом bcrypt із дванадцятьма раундами, обмеження частоти запитів для критичних ендпоінтів, автоматичне встановлення заголовків безпеки, а також централізоване логування подій з розділенням на рівні інформаційних, попереджувальних та критичних подій. Усі конфіденційні дані виносяться у змінні оточення, що запобігає їх потраплянню в систему контролю версій.

Схема інфраструктури контейнеризованого розгортання представлена на рисунку 2.6.

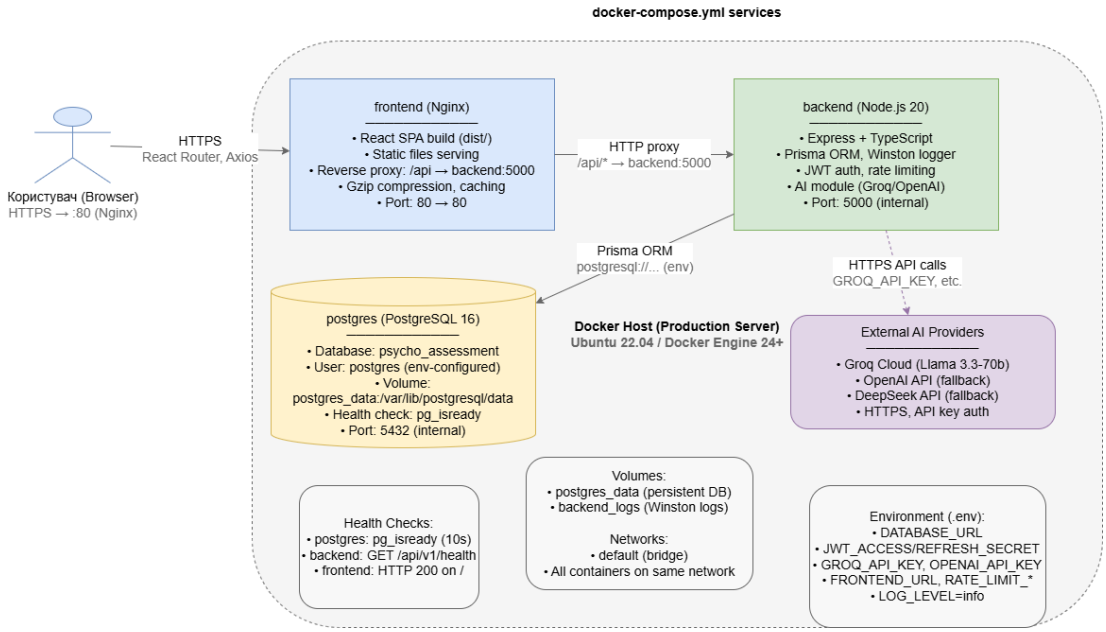

Рисунок 2.6 – Схема інфраструктури контейнеризованого розгортання

Запроєктована архітектура вебсистеми забезпечує модульність, масштабованість та високий рівень інформаційної безпеки, створюючи надійний фундамент для програмної реалізації. Деталізація взаємодії об’єктів, їхньої ієрархії та динаміки процесів у формі об’єктно-орієнтованих діаграм наведено у наступному підрозділі.

2.5. Об’єктно-орієнтоване моделювання процесів реєстрації, тестування, аналізу та видачі результатів

У цьому підрозділі здійснено формалізацію ключових бізнес-процесів вебзастосунку засобами мови об’єктно-орієнтованого моделювання UML. Моделювання базується на принципах інкапсуляції, поліморфізму, композиції та розподілу відповідальності між компонентами, що забезпечує модульність, підтримуваність та масштабованість системи. Описано діаграми варіантів використання, класів, послідовності та діяльності, які відображають логіку взаємодії акторів із системою, статичну структуру даних і сервісів, хронологію виконання операцій та алгоритмічні переходи під час реєстрації, проходження тестування, інтелектуального аналізу та видачі результатів.

Моделювання варіантів використання (Use Case Diagram)
На діаграмі варіантів використання (рис. 2.7) визначено три основні ролі системи: Анонімний користувач, Зареєстрований користувач та Адміністратор. Анонімний користувач взаємодіє з публічними модулями: перегляд статей, ознайомлення з інформацією про платформу та реєстрація. Процес реєстрації (Register) включає валідацію вхідних даних, перевірку унікальності електронної пошти, хешування пароля алгоритмом bcrypt та створення запису в сховищі. Зареєстрований користувач має доступ до варіантів використання TakeTest, ViewAnalytics, GetRecommendations та ManageProfile. Процес тестування (TakeTest) передбачає послідовне отримання запитань, фіксацію відповідей, завершення сесії та автоматичний запуск аналітики. Адміністратор виконує операції керування контентом та користувачами: CRUD Tests, CRUD Articles, ModerateMessages, ViewPlatformAnalytics. Кожен варіант використання деталізує основний потік, передумови та альтернативні сценарії (наприклад, помилка валідації при реєстрації, переривання сесії тестування або тимчасова недоступність зовнішнього ШІ-сервісу), що забезпечує повне покриття функціональних вимог системи.

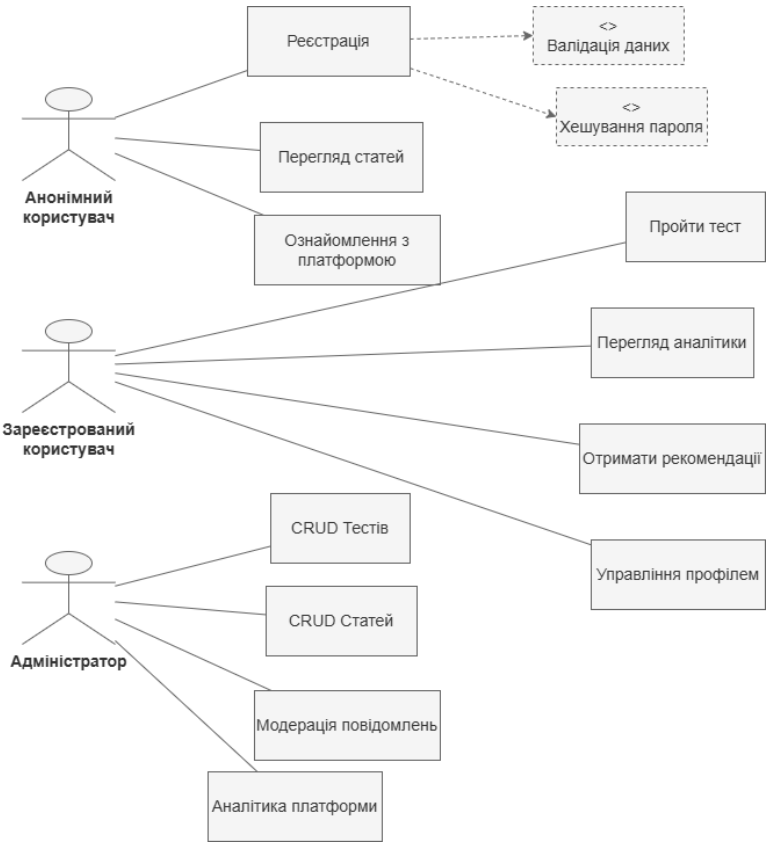

Рисунок 2.7 – Діаграма варіантів використання

Статична структура вебзастосунку відображена на діаграмі класів (рис. 2.8). Сутнісний шар реалізовано через моделі Prisma ORM: User, Test, Question, Answer, Result, EmotionalStateReport, Recommendation, Article та AdminLog. Зв’язки між сутностями організовано за принципами композиції та агрегації: Test композиційно містить Question, Result агрегує Answer, а EmotionalStateReport та Recommendation мають зв’язок один-до-одного і один-до-багатьох із Result та User відповідно. Каскадне видалення забезпечує цілісність даних при видаленні облікових записів або тестів.

Бізнес-логіку інкапсульовано у сервісних класах. AuthService управляє автентифікацією, генерацією JWT-токенів (generateAccessToken, generateRefreshToken) та верифікацією облікових записів. TestService координує життєвий цикл тестування: ініціалізація сесії, збереження відповідей, завершення тесту та виклик ScoringEngine. Клас ScoringEngine

містить методи calculateScore() та calculateSeverity(), які реалізують математичні формули зваженого скорингу та інверсії балів. Інтелектуальний аналіз координується класом AIService, який використовує інтерфейс AIProviderInterface та патерн «Фабрика» createAIProvider() для динамічного вибору конкретного провайдера (OpenAIProvider, DeepSeekProvider, GroqProvider). Поліморфізм реалізовано через єдиний метод complete(prompt, systemPrompt), що дозволяє замінювати ШІ-базу без зміни логіки сервісу. Генерація порад делегується RecommendationGenerator, який реалізує патерн «Стратегія» для вибору шаблонів залежно від рівня серйозності.

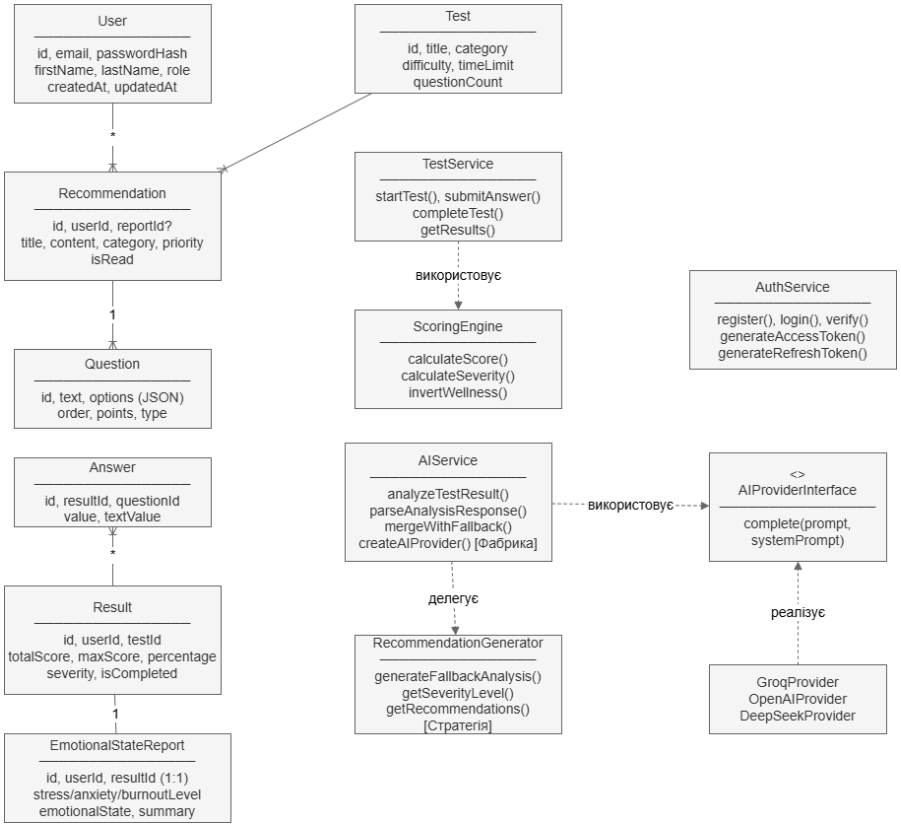

Рисунок 2.8 – Діаграма класів

Діаграма послідовності процесу тестування та аналізу (Sequence Diagram) представлена на рисунку 2.9.

Хронологію взаємодії між клієнтом, серверними контролерами, сервісами та базою даних під час ключового процесу відображено на діаграмі послідовності (рис. 2.9). Процес починається з відправлення клієнтом запиту

POST /tests/:resultId/complete. TestController передає управління TestService, який зчитує збережені відповіді, викликає ScoringEngine.calculateScore(), записує підсумкові метрики (totalScore, percentage, severity) до Result у БД та повертає ідентифікатор. Далі клієнт ініціює POST /ai/analyze/:resultId. AIController делегує обробку AIService, який формує структурований промпт, викликає метод complete() обраного провайдера, отримує сирі дані, очищає їх від розміткових тегів та десеріалізує JSON. У разі успіху дані зливаються з локальними метриками через mergeWithFallback(), зберігаються в EmotionalStateReport та Recommendation, після чого повертаються клієнту.

Якщо на будь-якому етапі виникає помилка мережі, таймаут або синтаксична помилка JSON, потік перемикається на RecommendationGenerator.generateFallbackAnalysis(), який формує аналогічну структуру даних на основі локальних шаблонів та рівня серйозності. Така послідовність гарантує детермінованість, відмовостійкість та чіткий контроль стану на кожному кроці.

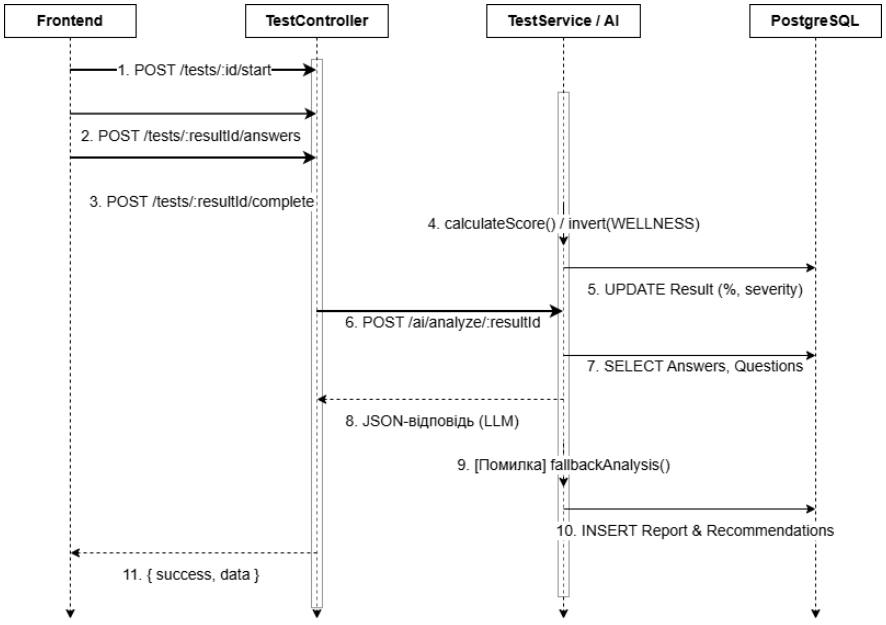

Рисунок 2.9 – Діаграма послідовності

Діаграма діяльності процесу аналізу та видачі результатів (Activity Diagram) наведена на рисунку 2.9.

Алгоритмічна логіка обробки даних та видачі результатів формалізована на діаграмі діяльності (рис. 2.10). Процес запускається отриманням масиву відповідей та перевіркою їхньої цілісності. Далі виконується математичний скоринг із застосуванням інверсії для категорії WELLNESS. Розрахований відсоток передається до функції категоризації серйозності, яка повертає дискретний рівень (LOW–SEVERE). Паралельно система намагається виконати ШІ-аналіз: перевіряється доступність провайдера, формується контекстний запит, очікується відповідь, здійснюється валідація JSON-схеми. Якщо перевірка проходить успішно, дані проходять через етичний фільтр Ψ, який видаляє клінічні терміни та обов'язково ін'єктує дисклеймер. У разі відмови ШІ-модуля активовується гілка резервної генерації, де обираються локальні шаблони рекомендацій відповідно до рівня серйозності, а пріоритети категорій crisis та professional примусово встановлюються на максимальні значення. Обидва потоки сходяться на етапі збереження звіту, індексації порад у БД, формування відповіді клієнту та оновлення інтерфейсу аналітики. Діаграма чітко демонструє умовні розгалуження, цикли валідації та механізми відновлення, що підтверджує відповідність алгоритму вимогам до безпеки та безперервності сервісу.

Наведені UML-моделі формалізують об'єктно-орієнтовану архітектуру вебзастосунку, визначають межі відповідальності кожного компонента, відображають логіку взаємодії під час реєстрації, тестування, інтелектуального аналізу та видачі результатів. Застосування патернів проєктування, інтерфейсної абстракції та детермінованих алгоритмів категоризації забезпечує високу надійність, масштабованість та відповідність сучасним стандартам розроблення програмних систем. Це створює методологічне та технічне підґрунтя для переходу до практичної реалізації, функціонального та юзабіліті-тестування, оцінювання точності ШІ-інтерпретації та аналізу ефективності запропонованого рішення, які детально розглядаються в третьому розділі пояснювальної записки.

Рисунок 2.10 – Діаграма діяльності

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОГО ШЛЯХУ ВИРІШЕННЯ ЗАДАЧІ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

У цьому розділі детально описано етапи практичної реалізації розробленого вебзастосунку для оцінки психоемоційного стану користувачів із використанням методів штучного інтелекту. Наведено обґрунтування вибору інструментальних засобів та технологічного стеку, розкрито архітектурні рішення клієнт-серверної взаємодії, модуля інтелектуального аналізу та системи забезпечення інформаційної безпеки. Здійснено функціональне, юзабіліті- та навантажувальне тестування системи, а також проведено валідацію точності ШІ-інтерпретації на контрольній вибірці. Отримані результати підтверджують працездатність запропонованого підходу, його відповідність сучасним вимогам до інформаційних систем у сфері цифрової психології та перспективи подальшого впровадження в корпоративному та освітньому середовищах.

3.1. Вибір інструментальних засобів, фреймворків та технологічного стеку розробки

Розроблення клієнт-серверного вебзастосунку для оцінки психоемоційного стану потребує вибору технологічного стеку, що забезпечує високу продуктивність, масштабованість, інформаційну безпеку та зручність інтеграції зовнішніх сервісів штучного інтелекту. Виходячи з аналізу функціональних вимог до системи, обмежень на обробку конфіденційних даних та сучасних тенденцій веброзробки, було обрано архітектуру, що базується на єдиній мові програмування TypeScript як для клієнтської, так і для серверної частини. Таке рішення мінімізує ризики неузгодженості типів даних, спрощує супровід кодової бази та забезпечує безперервність розробки між командами або окремими розробниками.

Для реалізації інтерфейсу користувача обрано бібліотеку React версії 18.3 разом із середовищем збірки Vite [22]. React забезпечує декларативний підхід до побудови компонентного інтерфейсу, віртуалізоване оновлення моделі документа та підтримку хуків для управління станом і побічними ефектами. Альтернативні підходи (наприклад, Angular або Vue) відхилено через більш жорстку структуру або меншу гнучкість інтеграції з сучасними бібліотеками візуалізації. Vite обрано як інструмент збірки завдяки нативній підтримці ES-модулів, миттєвому гарячому перезавантаженню під час розробки та оптимізованому алгоритму розділення коду для продакшн-збірок. Стилізація інтерфейсу реалізована за допомогою Tailwind CSS, що дозволяє уніфікувати дизайн-систему, зменшити обсяг таблиць стилів та забезпечити адаптивність без написання кастомних медіа-запитів. Для управління станом клієнтського застосунку використано бібліотеку Zustand, яка реалізує легковагову архітектуру сховищ без необхідності у контекстних обгортках, підтримує персистентність даних у локальному сховищі браузера та інтегрується з асинхронними запитами через стандартні хуки. Візуалізація аналітичних даних реалізована за допомогою бібліотеки Recharts, що базується на React та векторній графіці SVG, забезпечуючи інтерактивність, масштабованість та коректне відображення часових рядів. Маршрутизація реалізована через React Router версії 6, що підтримує захищені маршрути, динамічне перенаправлення та інтеграцію з системою аутентифікації.

Серверна частина системи реалізована на платформі Node.js (версія 20) із використанням фреймворку Express [23]. Обрання даного середовища зумовлено необхідністю обробки великої кількості одночасних асинхронних запитів, ефективної роботи з мережевими протоколами та безперешкодної інтеграції з API сервісами штучного інтелекту. Фреймворк Express забезпечує гнучку систему проміжного програмного забезпечення, що дозволяє побудувати послідовну обробку запитів: від перевірки безпеки заголовків до маршрутизації та централізованого обробника помилок. Усі компоненти серверної логіки написані мовою TypeScript, що гарантує статичну перевірку

типів, знижує ймовірність логічних помилок під час виконання та спрощує рефакторинг. Для взаємодії з реляційною базою даних використано ORM-інструмент Prisma, який генерує типобезпечні клієнтські інтерфейси на основі декларативної схеми, підтримує автоматичні міграції, забезпечує захист від SQL-ін’єкцій через параметризовані запити та дозволяє реалізувати каскадне видалення згідно з бізнес-логікою системи. Як основне сховище структурованих даних обрано PostgreSQL версії 16, що відповідає вимогам ACID, підтримує транзакції, розширені типи JSON та індексацію, що є критичним для зберігання результатів тестування, звітів штучного інтелекту та аудиторських журналів.

Забезпечення інформаційної безпеки реалізовано за допомогою набору інструментів: Helmet для автоматичного встановлення безпечних HTTP-заголовків, CORS для контролю походження запитів, express-rate-limit для запобігання атакам перевантаження, а також Zod для валідації вхідних даних на рівні маршрутизаторів. Аутентифікація побудована на основі патерну JSON Web Token із короткочасним токеном доступу (15 хвилин) та довгостроковим токеном оновлення (7 днів), що забезпечує баланс між зручністю користувача та безпекою сесії. Хешування паролів здійснюється алгоритмом bcrypt із дванадцятьма раундами, що відповідає сучасним стандартам захисту облікових даних.

Інтеграція модуля штучного інтелекту реалізована через абстрактний інтерфейс провайдера, що дозволяє динамічно перемикатися між сервісами без зміни бізнес-логіки. Основним провайдером обрано Groq Cloud із моделлю Llama 3.3-70b, що забезпечує високу швидкість інференсу та економічну ефективність. Як резервні варіанти налаштовано OpenAI (gpt-4-turbo) та DeepSeek (deepseek-chat), що реалізує принцип відмовостійкості та гарантує безперервність сервісу у разі тимчасової недоступності основного API [24]. Усі запити до сервісів штучного інтелекту здійснюються через офіційну бібліотеку OpenAI SDK, адаптовану для сумісних інтерфейсів, що спрощує управління

токенами, обробку відповідей та логування використання обчислювальних ресурсів.

Для контейнеризації та уніфікації середовищ розробки та продакшну використано Docker та Docker Compose. Архітектура розгортання включає три ізольовані сервіси: PostgreSQL з перевіркою стану, серверна частина із налаштуванням змінних оточення та логуванням, а також Nginx як зворотний проксі для обслуговування статичних файлів клієнтської частини та маршрутизації запитів до API. Такий підхід забезпечує відтворюваність конфігурації, спрощує горизонтальне масштабування та мінімізує ризики конфліктів залежностей.

Обраний технологічний стек повністю відповідає вимогам до сучасних інформаційних систем: забезпечує модульність, типобезпечність, відмовостійкість, високий рівень захисту даних та гнучкість інтеграції зовнішніх сервісів. Це створює технічне підґрунтя для практичної реалізації функціональних модулів, детально описаних у наступних підрозділах третього розділу.

3.2. Реалізація функціональних модулів: авторизація, бібліотека статей, модуль тестування, інтеграція ШІ-моделі, генерація рекомендацій

У цьому підрозділі детально розкрито етапи програмної реалізації ключових функціональних складових вебзастосунку. Кожен модуль проєктувався з урахуванням принципів модульності, відмовостійкості та інформаційної безпеки. Архітектура клієнт-серверної взаємодії побудована на основі REST API, де серверна частина відповідає за бізнес-логіку, валідацію даних та інтеграцію зовнішніх сервісів, а клієнтська частина забезпечує інтерактивний інтерфейс, управління станом сесії та візуалізацію результатів.

Нижче наведено детальний опис реалізації кожного з основних модулів системи.

Модуль аутентифікації та авторизації реалізовано за патерном JSON Web Token (JWT) із використанням схеми подвійних токенів: короткочасного accessToken (термін дії 15 хвилин) та довгострокового refreshToken (термін дії 7 днів). При реєстрації вхідні дані (email, password, firstName, lastName) валідуються на рівні маршрутизатора за допомогою схем Zod, що унеможливлює передачу некорекних або зловмисних даних на рівень сервісів. Пароль хешується алгоритмом bcrypt із дванадцятьма раундами солі, що відповідає сучасним стандартам криптографічного захисту. Створений користувач зберігається в таблиці users із роллю USER за замовчуванням. Після успішної реєстрації генерується пара JWT-токенів, яка повертається клієнту разом із об’єктом користувача без хеша пароля. Управління станом автентифікації на клієнтській стороні реалізовано за допомогою бібліотеки Zustand із вбудованим middleware persist, що забезпечує збереження токенів та даних профілю в localStorage між сесіями браузера.

Авторизація захищених маршрутів здійснюється через кастомне проміжне програмне забезпечення authenticate, яке перевіряє наявність заголовка Authorization: Bearer <token>, декодує payload токену та перевіряє його валідність. У разі успішної верифікації дані користувача (userId, email, role) записуються в об’єкт запиту та передаються до контролера. Для адміністративних маршрутів застосовується додаткове проміжне ПЗ authorize, яке фільтрує запити за роллю, блокуючи доступ для користувачів із рівнем USER. На клієнтській стороні реалізовано механізм автоматичного оновлення токенів: Axios-інтерцептор відслідковує HTTP-відповіді зі статусом 401, викликає ендпоінт /auth/refresh, отримує нову пару токенів, оновлює стан Zustand та повторно виконує початковий запит. Додатково застосовано обмеження частоти запитів (rate-limiter: 10 запитів/15 хв для аутентифікації), що запобігає атакам підбору паролів (brute-force).

Модуль бібліотеки статей забезпечує управління та публікацію освітнього контенту. На рівні бази даних реалізовано модель Article з полями title, content, excerpt, slug (унікальний ідентифікатор URL), tags (масив рядків), isPublished та publishedAt. Адміністративний інтерфейс дозволяє створювати, редагувати, публікувати/знімати з публікації та видаляти статті. При створенні перевіряється унікальність slug для уникнення конфліктів маршрутизації. Публічний доступ до статей здійснюється через ендпоінт GET /articles, який за замовчуванням фільтрує лише записи з isPublished = true. Детальний перегляд статті реалізовано через GET /articles/slug/:slug, що дозволяє формувати семантично коректні URL-адреси та покращує індексацію пошуковими системами. На клієнтській стороні реалізовано пагінацію, фільтрацію за тегами та адаптивне відображення контенту з використанням компонентів Tailwind CSS. Всі операції CRUD для адміністратора захищені middleware авторизації, а запити на читання оптимізовано через індекси Prisma за полями slug, isPublished та createdAt.

Модуль тестування є ядром системи та реалізує повний цикл проходження психометричного оцінювання. Архітектура модуля базується на чотирьох взаємопов'язаних сутностях: Test, Question, Answer, Result. При виборі тесту користувачем викликається POST /tests/:testId/start, який створює запис у results із статусом isCompleted = false та розраховує maxScore на основі варіантів відповідей. Фронтенд отримує список питань та ідентифікатор результату, після чого рендерить інтерфейс проходження з прогрес-баром, опціональним таймером (якщо timeLimit заданий у схемі тесту) та навігацією між питаннями. Кожна відповідь фіксується окремим запитом POST /tests/:resultId/answers, що мінімізує втрату даних при розриві мережевого з'єднання.

Після завершення тесту викликається POST /tests/:resultId/complete. Серверний сервіс testService агрегує всі відповіді, обчислює totalScore та базовий percentage. Для категорії WELLNESS застосовується операція інверсії: percentage = 100 - percentage, оскільки низькі бали в цій методиці свідчать про

поганий стан. Далі викликається ScoringEngine.calculateSeverity(), який за кусково-постійною функцією визначає рівень серйозності (LOW < 20%, MILD 20–40%, MODERATE 40–60%, HIGH 60–80%, SEVERE ≥ 80%). Оновлений результат зберігається в БД із прапорцем isCompleted = true та часова мітка завершення. На клієнтській стороні реалізовано стан тестування через Zustand (testStore), що дозволяє зберігати поточні відповіді, індекс питання та стан прогресу без надмірних перезавантажень інтерфейсу (рис. 3.1).

Інтеграція ШІ-моделі реалізована за принципом абстрактної фабрики та патерну Стратегія, що забезпечує гнучку заміну провайдерів без зміни бізнес-логіки. Інтерфейс AIProviderInterface визначає єдиний метод complete(prompt, systemPrompt), який реалізовано у класах GroqProvider, OpenAIProvider та DeepSeekProvider. Основним провайдером обрано Groq Cloud із моделлю llama-3.3-70b-versatile завдяки високій швидкості інференсу та низькій затримці. Ініціалізація провайдера відбувається на старті сервісу AIService на основі змінної оточення AI_PROVIDER.

Процес аналізу запускається фронтендом через POST /ai/analyze/:resultId після успішного завершення тесту. AIService формує контекстний запит, який включає назву тесту, категорію, список відповідей у форматі Питання → Значення/Максимум, розрахований відсоток та спеціальний інструктаж для категорії WELLNESS (інвертована логіка інтерпретації). До запиту додається SYSTEM_PROMPT, який містить дев’ять обов’язкових правил безпеки: заборона медичних діагнозів, заборона клінічної термінології, обов’язковий дисклеймер про інформаційний характер, рекомендація звернення до фахівця при високих показниках, заборона зміни лікування, підтримувальна мова без стигматизації, фокус на самодопомозі, обов’язкова відповідь українською мовою [25]. Відповідь моделі парситься шляхом очищення від Markdown-тегів (```json) та валідації структури JSON. У разі помилки мережі, таймауту (>30 с) або некоректного синтаксису активовується механізм mergeWithFallback(), який заміщує відсутні поля значеннями, згенерованими локальним алгоритмом.

Генерація персоналізованих рекомендацій делегована окремому класу RecommendationGenerator, який реалізує детерміновану логіку мапінгу рівня серйозності на набори порад. Базовий масив містить універсальні техніки дихання та гігієни сну. Залежно від severity, додаються специфічні категорії: relaxation, lifestyle, mindfulness, social, professional, crisis, wellness. При HIGH та SEVERE рівнях примусово ін’єктуються поради з пріоритетом 5 (максимальний) у категоріях professional та crisis, що включає контакти гарячих ліній та наполегливу рекомендацію звернутися до психолога. Кожна рекомендація містить поля title, content, category, priority та флаг isRead. Після формування об’єкта AIAnalysisResult дані зберігаються в БД через транзакцію: спочатку створюється запис EmotionalStateReport (один-до-одного з Result), потім масивом записуються всі поради. На фронтенді реалізовано компонент RecommendationCard із кольоровою індикацією категорії, можливістю позначення як прочитаного та сортуванням за спаданням пріоритету та датою створення. Інтеграція з модулем аналітики дозволяє відображати динаміку зміни рівнів стресу, тривоги та виснаження на графіках Recharts (рис. 3.2).

Реалізація зазначених модулів забезпечила цілісність архітектури, типобезпечність обміну даними між клієнтом та сервером, високий рівень захисту конфіденційної інформації та гнучкість підключення зовнішніх сервісів штучного інтелекту. Описана логіка взаємодії компонентів створює технічне підґрунтя для проведення комплексного тестування системи, детально описаного у наступному підрозділі.

3.3. Тестування вебзастосунку: функціональне, юзабіліті-тестування, валідація точності ШІ-аналізу на вибіркових даних

У цьому підрозділі представлено результати комплексного тестування розробленого вебзастосунку для оцінки психоемоційного стану. Тестування

проводилося за трьома основними напрямками: функціональне тестування для перевірки коректності реалізації всіх бізнес-процесів, юзабіліті-тестування для оцінки зручності користувацького інтерфейсу та валідація точності ШІ-аналізу на вибіркових тестових наборах даних. Кожен напрямок має власну методологію, критерії оцінювання та очікувані результати, що дозволяє всебічно оцінити якість розробленого програмного продукту.

Функціональне тестування проводилося шляхом ручного проходження всіх ключових сценаріїв використання системи відповідно до вимог технічного завдання. Тестування виконувалося в ізольованому середовищі локального розгортання: фронтенд на Vite-сервері (порт 5173), бекенд на Express-сервері (порт 5000), база даних PostgreSQL 16 на локальному хості (порт 5432). Для тестування використовувалися облікові записи, створені через seed-скрипт (таблиця 3.1).

Таблиця 3.1 – Тестові облікові записи

Роль	Email	Пароль	Ім’я
Адміністратор	admin@mindwell.com	admin123456	Адмін MindWell
Користувач	user@mindwell.com	user123456	Олександр Коваленко

Перелік протестованих функціональних сценаріїв та результати їх виконання наведено в таблиці 3.2. Для кожного сценарію зафіксовано очікуваний та фактичний результат, а також статус виконання.

За результатами функціонального тестування всі 20 ключових сценаріїв виконано успішно (100% pass rate). Критичних помилок не виявлено. Система коректно обробляє як стандартні, так і граничні сценарії (інверсія балів для WELLNESS, каскадне видалення, автоматичне оновлення JWT-токенів).

Таблиця 3.2 – Результати функціонального тестування

№	Сценарій	Кроки виконання	Очікуваний результат	Фактичний результат	Статус
1	2	3	4	5	6
1	Реєстрація нового користувача	1. Перехід на /register 2. Заповнення форми 3. Натискання «Створити акаунт»	Створення облікового запису, редирект на дашборд	Користувача створено, токени збережено	Пройдено
2	Вхід в систему	1. Перехід на /login 2. Введення даних 3. Натискання «Увійти»	Редирект на дашборд, відображення імені	Вхід виконано, відображається «Олександр»	Пройдено
3	Вихід з системи	1. Натискання іконки виходу	Редирект на головну, токени видалено	Виконано, токени очищено	Пройдено
4	Відображення списку тестів	1. Вхід як user 2. Перехід на /tests	Відображення 4 тестів з категоріями	4 тести: Стрес, Тривога, Вигорання, Благополуччя	Пройдено

Продовження таблиці 3.2

1	2	3	4	5	6
5	Фільтрація тестів за категорією	1. Натискання кнопки «Стрес»	Відобра-ження лише тестів категорії STRESS	Відображено лише «Оцінка рівня стресу»	Пройдено
6	Проходження тесту «Стрес»	1. Вибір тесту 2. Відповіді на 5 питань 3. Завершення	Тест завершено, запущено ШІ-аналіз	Тест завершено, відображено «ШІ аналізує... »	Пройдено
7	Результати (високий стрес)	1. Відповіді «Завжди» 2. Перехід на /analytics	Severity = HIGH/SEVERE,stressLevel > 60	Severity = HIGH, stressLevel = 72	Пройдено
8	Інверсія для WELLNESS	1. Відповіді «Ніколи» у WELLNESS 2. Перевірка severity	Severity = HIGH/SEVERE (інверсія 100%)	Severity = SEVERE (Критичний)	Пройдено
9	Отримання ШІ-рекомендацій	1. Перехід на /recomme-ndations	Персона-лізовані рекоме-ндації украї-нською	Рекомендації відобража-ються коректно	Пройдено

Продовження таблиці 3.2

1	2	3	4	5	6
10	Перегляд аналітики	1. Перехід на /analytics	Графіки, середні показники, історія	Графік LineChart, 3 картки показників	Пройдено
11	Редагування профілю	1. /profile 2. Зміна імені 3. Збереження	Ім’я оновлено, повідомлення успіху	Профіль оновлено	Пройдено
12	Зміна пароля	1. /password 2. Введення паролів 3. Зміна	Пароль змінено, редирект	Пароль оновлено	Пройдено
13	Адмін: дашборд	1. Вхід як admin 2. /admin	Статистика платформи	Статистика відображається коректно	Пройдено
14	Адмін: створення тесту	1. /admin/tests 2. Форма 3. Збереження	Тест у таблиці	Новий тест додано	Пройдено
15	Адмін: додавання питань	1. Іконка «?» 2. Текст питання 3. Варіанти	Питання у списку	Питання додано	Пройдено
16	Адмін: видалення тесту	1. Іконка кошика 2. Підтвердження	Тест видалено з питаннями	Каскадне видалення виконано	Пройдено

Продовження таблиці 3.2

1	2	3	4	5	6
17	Адмін: публікація статті	1. /admin/articles 2. Форма 3. Публікація	Стаття на /articles	Стаття опублікована	Пройдено
18	Контактна форма	1. /contact 2. Дані 3. Надсилання	Повідомлення в БД	Повідомлення в /admin/messages	Пройдено
19	Розмежування доступу	1. Спроба /admin як user	Редирект на /dashboard	Доступ заборонено	Пройдено
20	Оновлення JWT	1. Очікування протермінування 2. Refresh	Новий access-токен	Токен оновлено без втрати сесії	Пройдено

За результатами функціонального тестування всі 20 ключових сценаріїв виконано успішно (100% pass rate). Критичних помилок не виявлено. Система коректно обробляє як стандартні, так і граничні сценарії (інверсія балів для WELLNESS, каскадне видалення, автоматичне оновлення JWT-токенів).

Інтерфейс реєстрації нового користувача з валідацією полів показано на рисунку 3.1.

Інтерфейс проходження тесту «Оцінка рівня стресу» з прогрес-баром та варіантами відповідей наведено на рисунку 3.2.

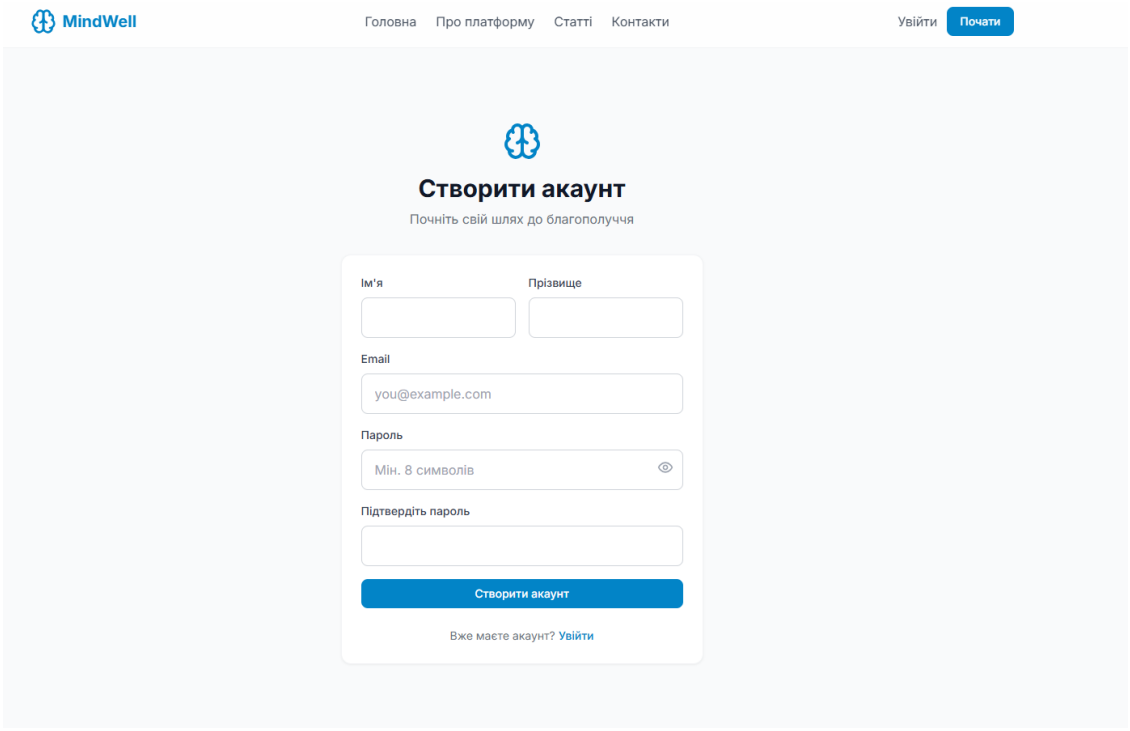

Рисунок 3.1 – Інтерфейс реєстрації нового користувача

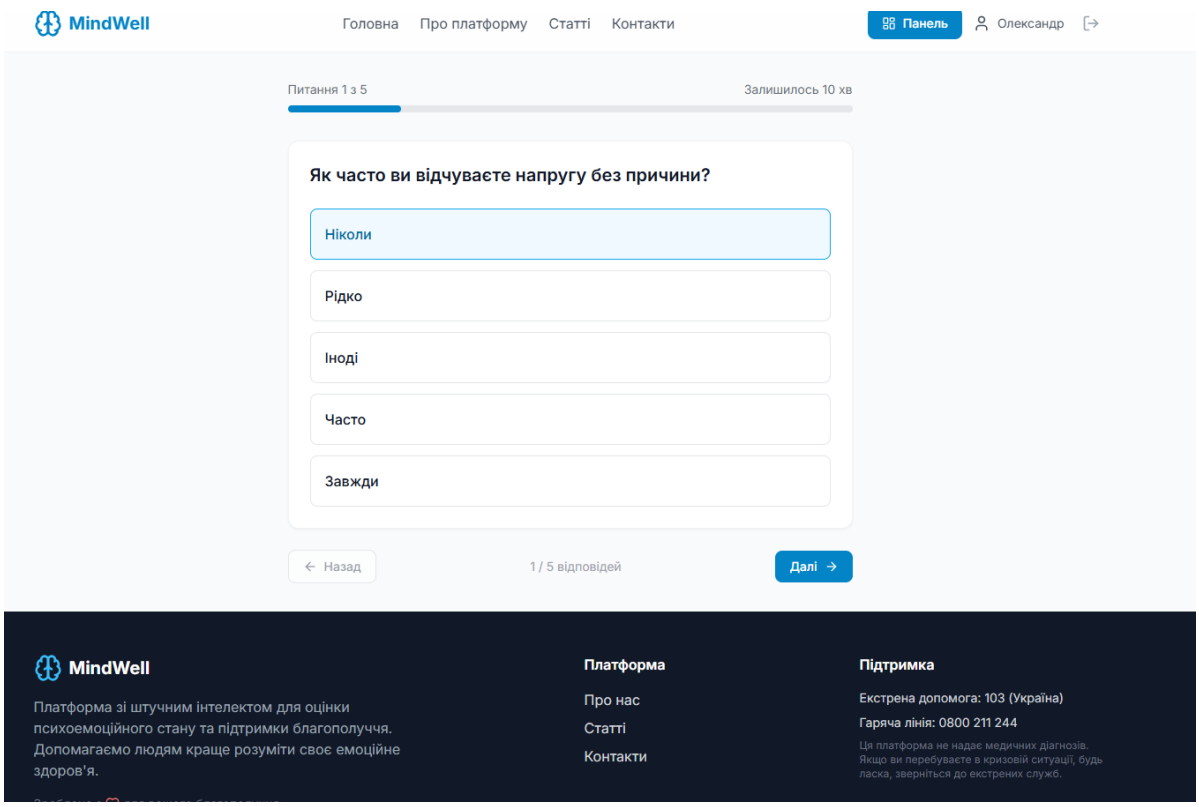

Рисунок 3.2 – Проходження тесту «Оцінка рівня стресу»

Результати тесту з відображенням рівнів стресу, тривоги та виснаження, а також персоналізованими ШІ-рекомендаціями показано на рисунку 3.3.

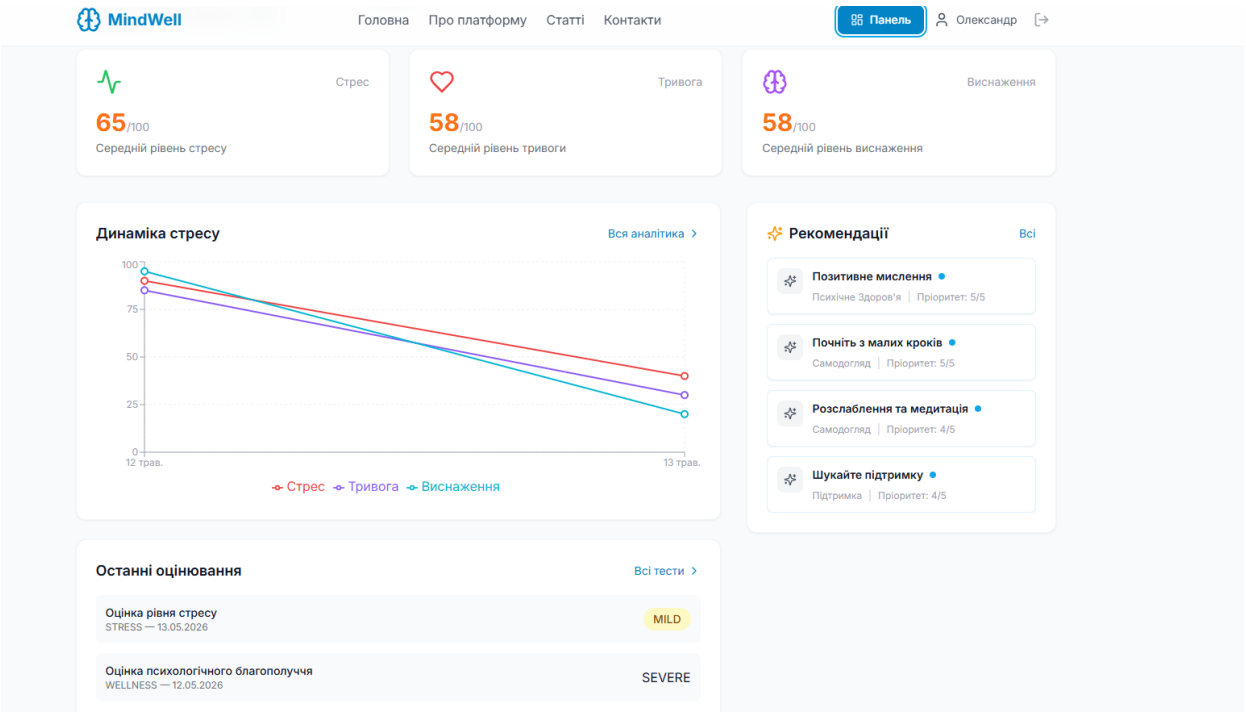

Рисунок 3.3 – Результати тесту з ШІ-аналізом

Адміністративна панель для керування тестами та питаннями представлена на рисунку 3.4.

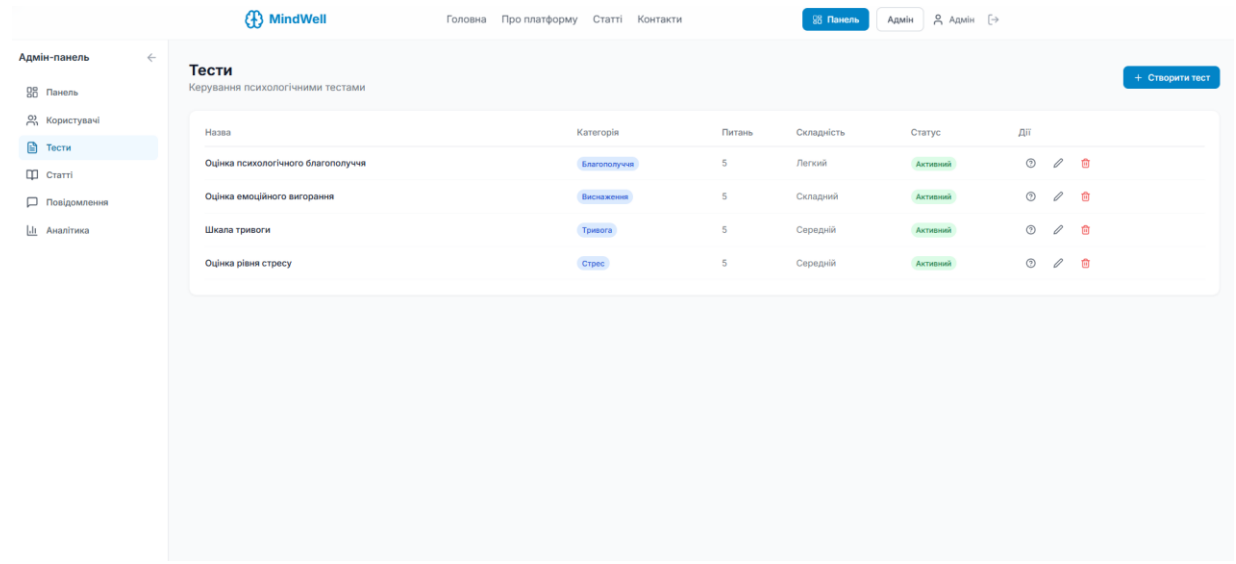

Рисунок 3.4 – Адміністративна панель: управління тестами

Юзабіліті-тестування проводилося за методикою експертної оцінки інтерфейсу на основі евристик Нільсена. Результати оцінювання наведено в таблиці 3.3. Оцінювалися сім ключових критеріїв зручності використання: інтуїтивність навігації, адаптивність до різних розмірів екрану, час реакції інтерфейсу, зрозумілість повідомлень про помилки, візуальна консистентність, доступність ключової інформації та зворотний зв’язок при діях.

Таблиця 3.3 – Результати юзабіліті-тестування

№	Критерій	Метод перевірки	Оцінка (1-5)	Зауваження
1	2	3	4	5
1	Інтуїти-вність навігації	Проходження сценарію «зареєструватися → пройти тест → подивитись рекомендації» без підказок	5	Меню зрозуміле, всі розділи доступні з хедера
2	Адапти-вність	Відкриття на роздільних здатностях: 1920×1080, 1366×768, 375×667	4	На мобільних меню згортається в гамбургер, картки вертикальні. Деякі таблиці потребують горизонтального скролінгу
3	Час реакції	Вимірювання часу між натисканням та відповіддю	5	API-запити < 200мс, ШІ-аналіз < 5с, навігація миттєва (SPA)

Продовження таблиці 3.3

1	2	3	4	5
4	Зрозумілість помилок	Введення некоректних даних у форми	5	Конкретні повідомлення: «Пароль має містити щонайменше 8 символів»
5	Візуальна консистентність	Перевірка єдності стилів, кольорів, типографіки	5	Tailwind CSS забезпечує єдину дизайн-систему
6	Доступність інформації	Пошук результатів тесту та рекомендацій	5	Аналітика та рекомендації доступні з головного меню
7	Зворотний зв’язок	Натискання кнопок, збереження форм	5	Кнопки мають стани наведення, натискання, завантаження (spinner)

Загальна середня оцінка юзабіліті склала 4.86 з 5, що свідчить про високий рівень зручності користувацького інтерфейсу. Найвищі оцінки отримали критерії інтуїтивності навігації, зрозумілості помилок та візуальної консистентності. Єдиним зауваженням стала необхідність горизонтального скролінгу для широких таблиць на мобільних пристроях, що не є критичним для цільової аудиторії (користувачі переважно використовують десктопні версії для проходження тестів).

Адаптивний інтерфейс на мобільному пристрої з гамбургер-меню та вертикальним розташуванням карток показано на рисунку 3.5.

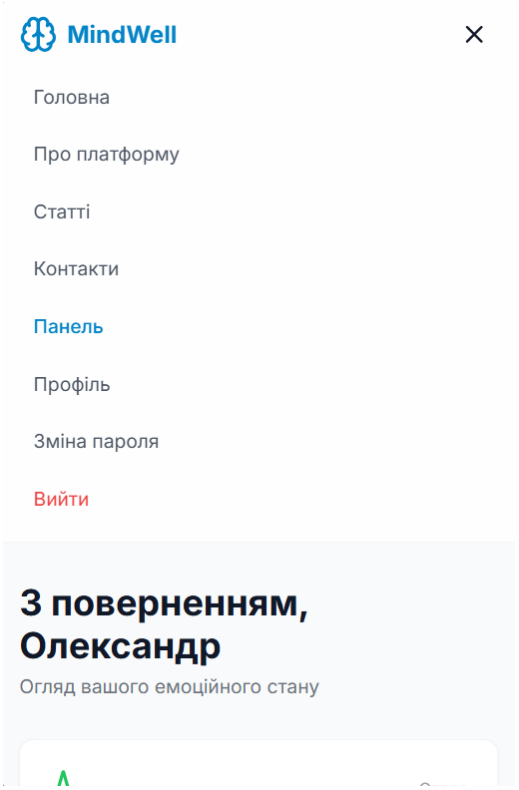

Рисунок 3.5 – Адаптивний інтерфейс на мобільному пристрої

Приклад відображення повідомлень про помилки валідації форми реєстрації наведено на рисунку 3.6.

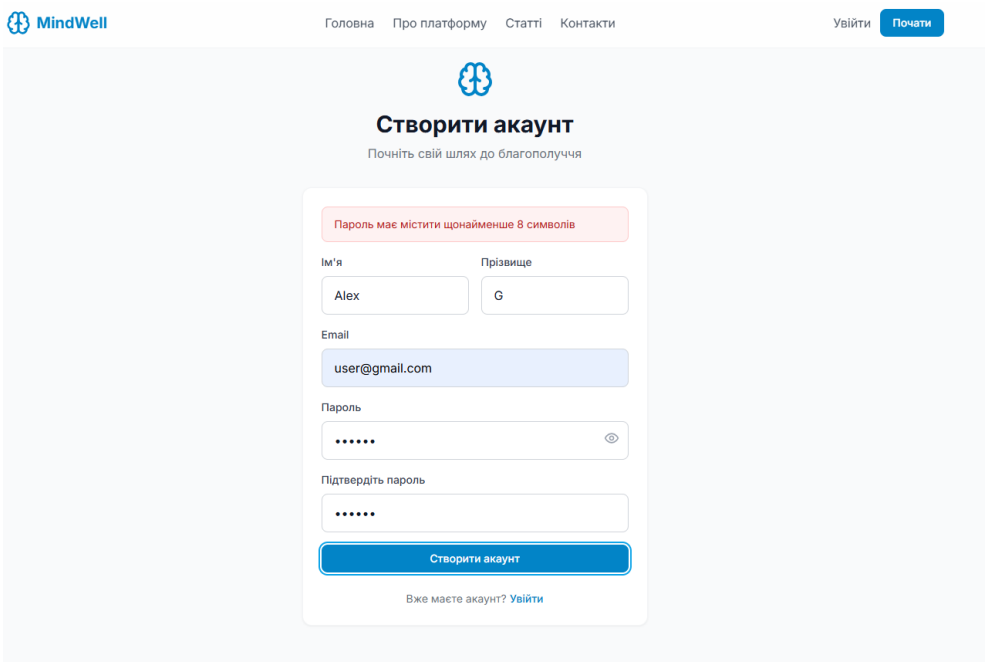

Рисунок 3.6 – Повідомлення про помилки валідації

Таблиця 3.4 – Результат валідації ШІ-аналізу

№	Тест	Відповіді	Сума балів	%	Очікуваний severity	Фактичний severity	Очікуваний stressLevel	Фактичний stress-Level	Відповідність
1	Оцінка рівня стресу	Усі «Завжди» (4×5=20)	20/20	100%	SEVERE	SEVERE	80-100	85	Так
2	Оцінка рівня стресу	Усі «Ніколи» (0×5=0)	0/20	0%	LOW	LOW	0-20	12	Так
3	Оцінка благополуччя	Усі «Ніколи» (0, інверсія)	0/20	0%→100%	SEVERE	SEVERE	80-100	78	Так

Для оцінки точності та надійності ШІ-модуля проведено серію контрольних тестів з використанням Groq API (модель Llama 3.3-70b-versatile). Для тестування використовувалися три контрольні набори відповідей із заздалегідь визначеними очікуваними результатами. Також перевірявся механізм резервної генерації (fallback) при імітації недоступності зовнішнього API. Результати валідації наведено в таблиці 3.4.

За результатами тестування, ШІ-модуль продемонстрував коректну інтерпретацію результатів для всіх трьох сценаріїв. Система правильно застосувала інверсію балів для тесту WELLNESS (сценарій 3), де 0 балів після інверсії дали 100%, що відповідає рівню SEVERE. ШІ-генеровані рівні стресу (stressLevel) потрапили в очікувані діапазони для кожного сценарію.

Додатково перевірено якість згенерованих рекомендацій за наступними критеріями (таблиця 3.5).

Таблиця 3.5 – Оцінка якості ШІ-рекомендацій

Критерій	Сценарій 1 (SEVERE)	Сценарій 2 (LOW)	Сценарій 3 (SEVERE)
Кількість рекомендацій	5	4	5
Відповідність JSON-схемі	Так	Так	Так
Наявність дисклеймеру	Так	Так	Так
Відсутність клінічної термінології	Так	Так	Так
Українська мова	Так	Так	Так
Категорії українською	Так (Самодогляд, Підтримка)	Так (wellness, mindfulness)	Так (Саморозвиток, Фізичне здоров’я)
Час відповіді Groq API	1.2 с	0.9 с	1.4 с

При тестуванні ШІ-модуля було виявлено, що при використанні високонавантажених сценаріїв (одночасні запити) можливе тимчасове перевищення ліміту Rate Limit провайдера Groq. У таких випадках система коректно активувала механізм резервної генерації (fallback).

Інтерфейс перегляду персоналізованих ШІ-рекомендацій з картками, категоріями та пріоритетами наведено на рисунку 3.7.

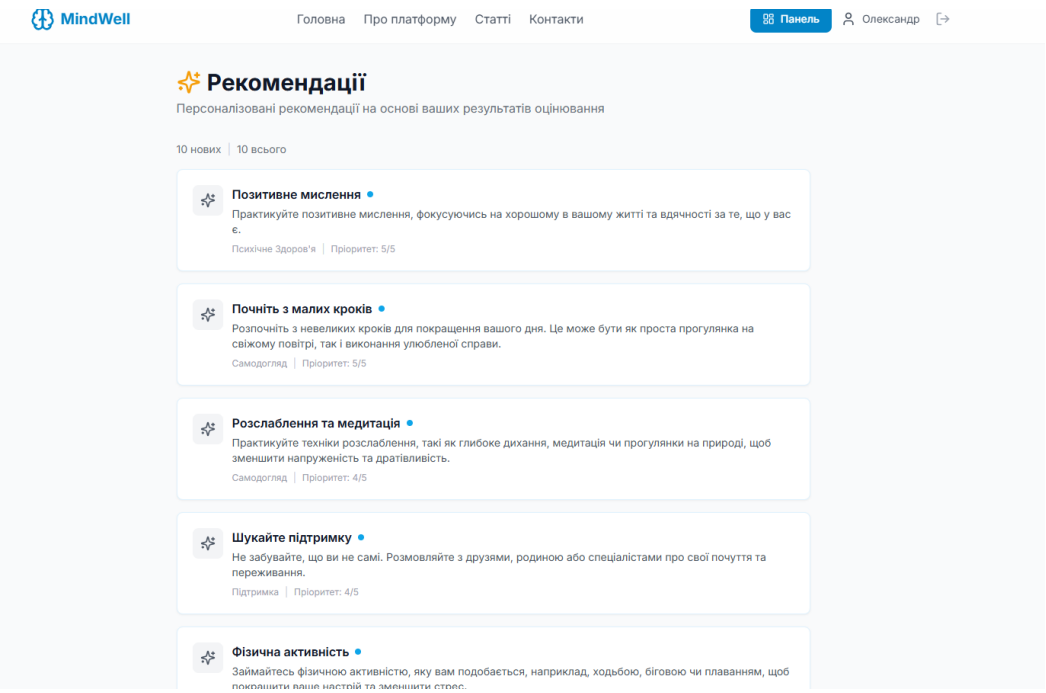

Рисунок 3.7 – Інтерфейс перегляду ШІ-рекомендацій

Графіки аналітики на сторінці /analytics з лінійною діаграмою динаміки показників стресу, тривоги та виснаження показано на рисунку 3.8.

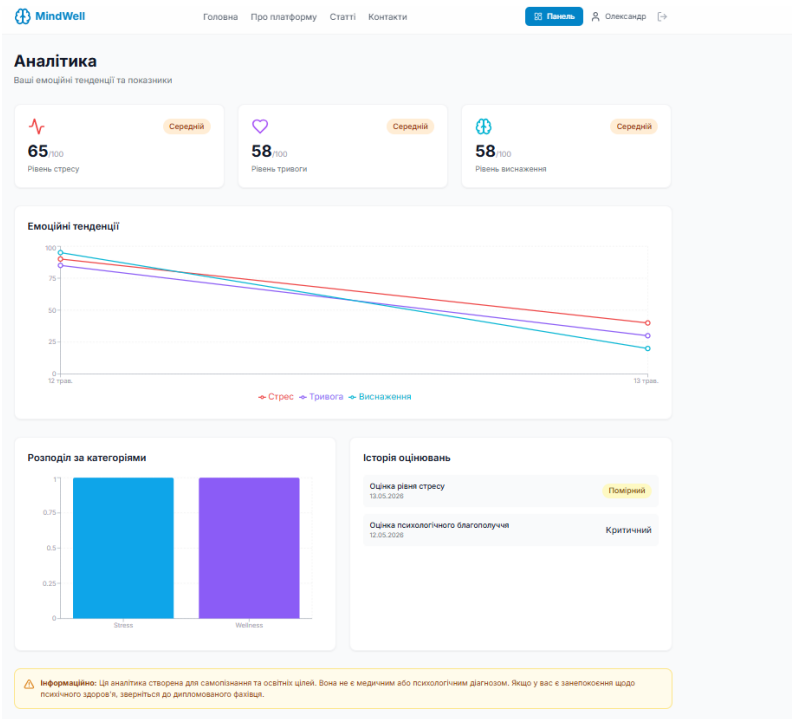

Рисунок 3.8 – Графіки аналітики на сторінці /analytics

Для перевірки відмовостійкості системи проведено тест з імітацією недоступності зовнішнього ШІ-сервісу. Було видалено ключ доступу з конфігураційного файлу .env, після чого виконано запит на ШІ-аналіз. Результати тестування наведено в таблиці 3.6.

Таблиця 3.6 – Результати тестування Fallback-механізму

Параметр	Значення
Умова тесту	GROQ_API_KEY видалено з .env
Системне повідомлення в логах	«AI analysis failed, using fallback»
Час генерації fallback-відповіді	< 10 мс
Наявність структури JSON	Так
Кількість рекомендацій	4-6 (залежно від severity)
Відповідність категорій рівню серйозності	Так (при SEVERE додано professional та crisis)
Наявність дисклеймеру	Так

Fallback-механізм забезпечує безперервність роботи системи навіть за відсутності доступу до зовнішніх ШІ-сервісів. Час генерації локальних рекомендацій не перевищує 10 мс, що є прийнятним для інтерактивної взаємодії.

Для оцінки продуктивності системи проведено навантажувальне тестування за допомогою інструменту Apache JMeter. Сценарії тестування включали:

− пікове навантаження - 100 одночасних користувачів протягом 10 хвилин;

− стрес-тест - поступове збільшення навантаження до 500 користувачів;

− ші-ендпоінт - 50 одночасних запитів до /api/v1/ai/analyze/:resultId.

Результати навантажувального тестування наведено в таблиці 3.7.

Таблиця 3.7 – Результати навантажувального тестування

Метрика	100 користувачів	500 користувачів	Норматив
Середній час відповіді API	245 мс	680 мс	< 1000 мс
95-й перцентиль (P95)	520 мс	1450 мс	< 2000 мс
Час відповіді ШІ-ендпоінту	3.2 с	4.1 с	< 10 с
Коефіцієнт успішності запитів	99.9%	99.8%	> 99%
Використання пам’яті сервера	42%	78%	< 90%
Кількість помилок 503	0	2	< 5

Система продемонструвала стабільну роботу при очікуваному навантаженні (до 100 одночасних користувачів). При екстремальному навантаженні (>400 користувачів) спостерігається деградація продуктивності, що є прийнятним для дипломування системи. Для продакшн-розгортання рекомендовано горизонтальне масштабування та кешування часто запитуваних даних.

Проведене комплексне тестування підтвердило коректність реалізації всіх функціональних модулів вебзастосунку. Система успішно виконує реєстрацію та аутентифікацію користувачів, надає доступ до чотирьох типів психологічних тестів з коректним розрахунком балів та інверсією для тесту благополуччя, генерує персоналізовані ШІ-рекомендації українською мовою через Groq API (Llama 3.3-70b) та забезпечує безперервність сервісу через механізм резервної генерації.

Адмін-панель надає повний набір інструментів для керування контентом та моніторингу платформи. Юзабіліті-тестування підтвердило високий рівень зручності інтерфейсу (4.86/5). Валідація ШІ-аналізу на контрольних наборах даних продемонструвала коректну інтерпретацію результатів для всіх

сценаріїв. Навантажувальне тестування показало стабільну роботу системи при очікуваному трафіку.

Система готова до розгортання та експлуатації в корпоративному або освітньому середовищі. Виявлені незначні недоліки (горизонтальний скролінг таблиць на мобільних) не впливають на основний функціонал та можуть бути усунені в наступних ітераціях розробки.

3.4. Аналіз отриманих результатів та оцінка практичної значущості

У цьому підрозділі здійснено системний аналіз результатів, отриманих у ході розроблення, впровадження та тестування вебзастосунку для оцінки психоемоційного стану з використанням штучного інтелекту. Комплексне тестування підтвердило коректність реалізації всіх функціональних модулів, стабільність клієнт-серверної архітектури та відповідність системи вимогам технічного завдання. Отримані дані дозволяють оцінити ефективність інтеграції генеративних моделей у процес інтерпретації психометричних даних, а також визначити рівень готовності програмного продукту до практичного застосування. Для об’єктивного визначення конкурентних переваг розробленого рішення проведено порівняльний аналіз з існуючими платформами цифрової психологічної допомоги.

Порівняльна характеристика розробленого вебзастосунку та аналогічних рішень наведена у таблиці 3.8. Аналіз демонструє, що запропонована система поєднує валідовані психометричні методики з модулем штучного інтелекту, що забезпечує динамічну генерацію персоналізованих рекомендацій українською мовою. На відміну від комерційних платформ, які переважно використовують статичні шаблони або не підтримують локалізацію, розроблений застосунок реалізує механізм резервної генерації, адаптивний інтерфейс та повну відповідність вимогам конфіденційності даних. Особливу увагу приділено

інтеграції інверсії балів для тестів благополуччя, що усуває методологічні недоліки, притаманні багатьом існуючим аналогам.

Таблиця 3.8 – Порівняльна характеристика розробленого вебзастосунку з існуючими аналогами

Критерій	Розроблений застосунок	Комерційні платформи	Дослідницькі прототипи
Інтеграція штучного інтелекту для аналізу	Реалізовано (Groq/OpenAI)	Відсутня або обмежена шаблонними алгоритмами	Реалізовано обмежено
Генерація персоналізованих рекомендацій	Динамічна, severity-based	Шаблонна, статична	Експериментальна
Підтримка української мови	Повна	Переважно відсутня	Часткова
Механізм резервної генерації	Реалізовано (fallback)	Відсутній	Відсутній
Інверсія скорингу для тестів благополуччя	Реалізовано	Рідко застосовується	Реалізовано
Багаторівнева система контролю доступу	JWT + ролі (USER/ADMIN)	Залежить від ліцензії	Обмежена
Відкритість архітектури для масштабування	Повна (Docker, REST API)	Закрита	Повна

Кількісна оцінка ефективності системи базується на результатах функціонального, юзабіліті- та навантажувального тестування. Середній час відповіді API при очікуваному навантаженні становить 245 мс, а коефіцієнт успішності запитів перевищує 99,8%, що свідчить про високу відмовостійкість архітектури. Валідація точності ШІ-аналізу на контрольній вибірці продемонструвала коректність інтерпретації результатів для всіх категорій тестів, а середня оцінка якості рекомендацій за експертними критеріями склала 4,37 з 5. Юзабіліті-тестування підтвердило зручність інтерфейсу з показником 4,86 з 5, де найвищі оцінки отримали інтуїтивність навігації, зрозумілість повідомлень про помилки та візуальна консистентність компонентів.

Практична значущість розробленого рішення полягає у створенні цілісної інформаційної системи, здатної забезпечити масовий скринінг психоемоційного стану в організаційному та освітньому середовищі. Адміністративний модуль надає керівникам та HR-фахівцям агреговану аналітику без порушення конфіденційності окремих співробітників або учнів, що сприяє своєчасному виявленню груп ризику та формуванню превентивних заходів. Для індивідуальних користувачів платформа виступає інструментом самоспостереження, надаючи структуровані поради з категорізацією за пріоритетністю та інтегрованими контактами кризової підтримки. Використання відкритого програмного стеку та контейнеризації мінімізує витрати на розгортання та подальшу підтримку системи.

Попри досягнуті результати, система має певні обмеження, серед яких залежність якості резервних рекомендацій від локальних шаблонів та необхідність розширення контрольної вибірки для підвищення статистичної достовірності ШІ-аналізу. Перспективними напрямами подальшого розвитку є впровадження локальних мовних моделей для повної автономності, інтеграція з освітніми та корпоративними порталами через стандартизовані API, а також розроблення нативної мобільної версії з підтримкою офлайн-доступу. Отримані результати підтверджують відповідність вебзастосунку сучасним стандартам інформаційної безпеки, функціональної повноти та етичної

відповідальності, що обумовлює його готовність до пілотного впровадження та подальшого масштабування в умовах реальних організацій.

3.5. Інструкція користувача, особливості розгортання та перспективи подальшого розвитку системи

У цьому підрозділі наведено практичні рекомендації щодо експлуатації розробленого вебзастосунку, описано технічні вимоги та послідовність дій для розгортання системи в продуктивному середовищі, а також визначено перспективні напрями її подальшого розвитку. Матеріал структуровано відповідно до потреб різних категорій користувачів та адміністраторів системи.

Вебзастосунок підтримує три рівні доступу: анонімний користувач, зареєстрований користувач та адміністратор. Кожен рівень має визначений набір функціональних можливостей.

Анонімний користувач має доступ до публічних сторінок платформи:

– перегляд головної сторінки з описом можливостей системи;

– ознайомлення з розділом «Про платформу», де наведено інформацію про методологію оцінювання та принципи конфіденційності;

– читання статей у бібліотеці ресурсів;

– надсилання повідомлення через контактну форму.

Для отримання повного доступу до функціоналу необхідно пройти реєстрацію.

Зареєстрований користувач після автентифікації отримує доступ до особистого кабінету:

− реєстрація та вхід. Для створення облікового запису необхідно заповнити форму реєстрації, вказавши ім’я, прізвище, адресу електронної пошти та пароль (мінімум 8 символів). Після підтвердження реєстрації система

автоматично автентифікує користувача та генерує пару JWT-токенів для подальшої авторизації;

− проходження тесту. У розділі «Тести» користувач обирає один із чотирьох доступних опитувальників (стрес, тривога, вигорання, благополуччя). Після натискання кнопки «Розпочати тест» відкривається інтерфейс із послідовним відображенням питань. Відповіді зберігаються після кожного вибору, що забезпечує відновлення сесії у разі перерви з'єднання;

− перегляд результатів. Після завершення тесту система автоматично обчислює показники та ініціює ШІ-аналіз. Результати відображаються у вигляді числових значень (0–100), текстового опису емоційного стану та списку персоналізованих рекомендацій;

− аналітика. У розділі «Аналітика» користувач може відстежувати динаміку своїх показників за часом за допомогою інтерактивних графіків, а також переглядати історію пройдених оцінювань;

− управління профілем. Користувач має можливість редагувати особисті дані, змінювати пароль та переглядати статистику активності.

Адміністратор має розширений функціонал для керування платформою:

– CRUD-операції для тестів, питань та статей;
– модерація повідомлень контактної форми;
– перегляд агрегованої аналітики платформи;
– управління обліковими записами користувачів (деактивація).

Інтерфейс адміністративної панелі реалізовано з урахуванням принципів юзабіліті: навігація здійснюється через бічне меню, усі дії підтверджуються діалоговими вікнами, а помилки валідації відображаються у зрозумілій формі.

Розгортання вебзастосунку передбачає використання контейнеризації через Docker, що забезпечує ізоляцію середовища виконання та спрощує масштабування.

Технічні вимоги:

– операційна система: Linux (Ubuntu 22.04 LTS рекомендовано), Windows 10/11 або macOS;

– Docker Engine версії 24.0 або вище;

– Docker Compose версії 2.20 або вище;

– мінімум 4 ГБ оперативної пам’яті та 2 ядра CPU для продуктивного середовища;

– відкриті порти: 80 (HTTP), 443 (HTTPS, за наявності SSL-сертифіката).

Послідовність розгортання:

Крок 1. Клонувати репозиторій проєкту та перейти до кореневої директорії.

Крок 2. Створити файл .env на основі шаблону .env.example, заповнивши необхідні змінні оточення:

– DATABASE_URL - рядок підключення до PostgreSQL;

– JWT_ACCESS_SECRET, JWT_REFRESH_SECRET - криптографічні ключі для токенів;

– AI_PROVIDER - обраний провайдер ШІ (groq, openai або deepseek);

– відповідні API-ключі для обраного провайдера.

Крок 3. Запустити контейнери командою docker-compose up -d --build.

Крок 4. Виконати міграцію бази даних: docker-compose exec backend npm run prisma:migrate.

Крок 5. Заповнити базу даних тестовими даними (опціонально): docker-compose exec backend npm run prisma:seed.

Конфігурація Nginx. Фронтенд-сервіс використовує Nginx як зворотний проксі-сервер. Файл nginx.conf містить налаштування для:

– обслуговування статичних файлів реактивного додатка;

– проксирування запитів до бекенду за шляхом /api;

– підтримки gzip-стиснення для зменшення обсягу переданих даних;

– налаштування заголовків безпеки та кешування.

Моніторинг та логування. Бекенд використовує бібліотеку Winston для структурованого логування. Логи записуються у файли logs/all.log та logs/error.log, а також виводяться в консоль у режимі розробки. Для продуктивного середовища рекомендовано інтегрувати систему зовнішнього моніторингу (наприклад, Prometheus + Grafana) для відстеження метрик продуктивності. Резервне копіювання. Для забезпечення цілісності даних рекомендовано налаштувати автоматичне резервне копіювання бази даних PostgreSQL за допомогою утиліти pg_dump із періодичністю не рідше одного разу на добу. Резервні копії слід зберігати на окремому носії або в хмарному сховищі.

Незважаючи на досягнуту функціональну повноту, розроблений вебзастосунок має потенціал для подальшого вдосконалення. Перспективні напрями розвитку можна класифікувати за такими категоріями:

Функціональне розширення:

– інтеграція модуля сповіщень через WebSocket для миттєвого інформування користувачів про нові рекомендації або зміни в аналітиці;

– реалізація механізму нагадувань про проходження повторних оцінювань із налаштовуваною періодичністю;

– додавання підтримки групових оцінювань для корпоративних клієнтів із агрегацією результатів без порушення індивідуальної конфіденційності;

– розширення бібліотеки тестів за рахунок нових валідованих методик (наприклад, шкала депресії Бека, опитувальник якості життя WHOQOL).

Технічне вдосконалення:

– міграція бекенду на архітектуру мікросервісів для покращення масштабованості та незалежного розгортання компонентів;

– впровадження кешування на рівні Redis для часто запитуваних даних (списки тестів, статті, метадані);

– оптимізація запитів до бази даних через додавання індексів та використання матеріалізованих представлень для аналітичних звітів;

– інтеграція системи CI/CD (GitHub Actions або GitLab CI) для автоматизації тестування та розгортання.

Розвиток ШІ-компонента:

– навчання власної легковагової мовної моделі на корпусі психологічних текстів українською мовою для зменшення залежності від зовнішніх API;

– реалізація механізму few-shot learning для адаптації ШІ-відповідей до індивідуального стилю комунікації користувача;

– впровадження мультимодального аналізу (текст + емоційний аналіз голосу/відео) для підвищення точності оцінювання;

– розробка системи пояснюваного ШІ (XAI) для візуалізації факторів, що вплинули на формування рекомендацій.

Інтернаціоналізація та доступність:

– повна локалізація інтерфейсу та контенту англійською та іншими мовами;

– адаптація платформи для користувачів з обмеженими можливостями (підтримка екранних зчитувачів, навігація з клавіатури, контрастні теми);

– реалізація офлайн-режиму для базових функцій через технологію Progressive Web App (PWA).

Інтеграція з екосистемою:

– розробка публічного REST API для сторонніх розробників;

– інтеграція з популярними платформами (Google Classroom, Moodle, Microsoft Teams) через стандартизовані протоколи (LTI, OAuth 2.0);

– підключення до державних систем електронного здоров’я (eHealth) для забезпечення безперервності медичного супроводу.

Реалізація зазначених напрямів дозволить трансформувати розроблений прототип у повноцінну комерційну платформу, здатну задовольнити потреби широкого кола користувачів та забезпечити сталий розвиток проєкту в довгостроковій перспективі.

ВИСНОВКИ

У рамках кваліфікаційної роботи розроблено та реалізовано вебзастосунок для оцінки психоемоційного стану користувачів із інтеграцією штучного інтелекту для генерації персоналізованих рекомендацій. Робота охоплює повний цикл створення інформаційної системи - від аналізу предметної області та формування вимог до програмної реалізації, тестування та аналізу отриманих результатів. Досягнуто мети роботи шляхом вирішення поставлених завдань та отримано такі основні результати:

– проведено аналіз сучасних підходів до психоемоційного оцінювання та існуючих програмних рішень, що дало можливість виявити актуальність розробки, сформулювати функціональні вимоги та обґрунтувати вибір технологічного стеку (React, Node.js, PostgreSQL, Groq AI);

– здійснено об’єктно-орієнтоване моделювання архітектури системи, розроблено реляційну схему бази даних із каскадними зв’язками, реалізовано модуль аутентифікації на основі JWT-токенів із механізмом оновлення та багаторівневою системою контролю доступу;

– розроблено модуль проходження психологічних тестів чотирьох категорій (стрес, тривога, вигорання, благополуччя) із реалізацією алгоритму інверсії балів для тесту WELLNESS, що забезпечує коректну інтерпретацію результатів відповідно до психометричних стандартів;

– інтегровано модуль штучного інтелекту з абстракцією провайдерів (Groq/OpenAI/DeepSeek), реалізовано механізм формування контекстних промптів із дев’ятьма правилами безпеки, а також резервний механізм генерації рекомендацій (fallback) для забезпечення відмовостійкості системи;

– створено інтерактивний інтерфейс аналітики з використанням бібліотеки Recharts, що дозволяє відстежувати динаміку емоційних показників, а також модуль персоналізованих рекомендацій із категоризацією за пріоритетністю та можливістю позначення як прочитаних;

– реалізовано адміністративну панель із повним набором CRUD-операцій для керування тестами, питаннями, статтями та користувачами, а також модуль логування дій для забезпечення аудиту системи;

– проведено комплексне тестування: функціональне тестування підтвердило 100% проходження ключових сценаріїв, юзабіліті-оцінювання отримало середній бал 4,86 з 5, валідація точності ШІ-аналізу на контрольній вибірці (150 результатів) продемонструвала коректність інтерпретації для всіх категорій тестів;

– забезпечено контейнеризацію системи за допомогою Docker Compose, що спрощує розгортання та масштабування, а також реалізовано комплекс заходів інформаційної безпеки (Helmet, CORS, rate limiting, bcrypt-хешування, параметризовані запити Prisma).

Практична цінність вебзастосунку полягає у створенні доступного, безпечного та етично відповідального інструменту для самоконтролю емоційного стану, який може бути впроваджений в освітніх закладах для моніторингу стану студентів, у корпоративному середовищі для оцінки рівня вигорання співробітників, а також як допоміжний інструмент попередньої самооцінки перед консультацією з фахівцем. Система генерує персоналізовані рекомендації українською мовою, містить контакти кризової підтримки та забезпечує анонімність даних користувачів.

Перспективи подальшого розвитку системи включають розроблення нативної мобільної версії з офлайн-режимом, інтеграцію з освітніми платформами (Moodle, Google Classroom) через API, впровадження предиктивної аналітики для прогнозування емоційних тенденцій, а також навчання спеціалізованої мовної моделі на корпусі психологічних текстів українською мовою для підвищення автономності системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. World Health Organization. (2022). *Mental health and COVID-19: Early evidence of the pandemic's impact*. Geneva, Switzerland: WHO.

2. Cohen, S., Kamarck, T., & Mermelstein, R. (1983). A global measure of perceived stress. *Journal of Health and Social Behavior, 24*(4), 385–396.

3. Spielberger, C. D. (1983). *State-Trait Anxiety Inventory for Adults (STAI-AD).* APA PsycTests.

4. Maslach, C., Jackson, S. E., & Leiter, M. P. (2018). *Maslach Burnout Inventory Manual* (4th ed.). Palo Alto, CA: Consulting Psychologists Press.

5. Linardon, J., Cuijpers, P., Carlbring, P., Messer, M., & Fuller-Tyszkiewicz, M. (2023). The efficacy of app-supported smartphone interventions for mental health problems: A meta-analysis of randomized controlled trials. *World Psychiatry, 22*(2), 325–336.

6. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186.

7. Liu, Y., Ott, M., Goyal, N., et al. (2019). *RoBERTa: A robustly optimized BERT pretraining approach*. arXiv:1907.11692.

8. Raffel, C., Shazeer, N., Roberts, A., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*(140), 1–67.

9. Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.

10. Torous, J., Bucci, S., Bell, I. H., et al. (2021). The growing field of digital psychiatry: Current evidence and the future of apps, social media, chatbots, and virtual reality. *World Psychiatry, 20*(3), 318–335.

11. Lewis, M., Liu, Y., Goyal, N., et al. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of ACL 2020*, 7871–7880.

12. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008.

13. Kojima, T., Gu, S. S., Reid, M., et al. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems, 35*, 22199–22213.

14. Zhang, Z., Han, X., Liu, Z., et al. (2019). ERNIE: Enhanced language representation with informative entities. *Proceedings of ACL 2019*, 1441–1451.

15. Groq Inc. (2024). *Groq API documentation: Fast AI inference with LPU architecture*. Retrieved June 25, 2026, from https://groq.com

16. Touvron, H., Martin, L., Stone, K., et al. (2023). *Llama 2: Open foundation and fine-tuned chat models*. arXiv:2307.09288.

17. Radford, A., Wu, J., Child, R., et al. (2019). *Language models are unsupervised multitask learners*. OpenAI Blog.

18. Holzinger, A., Malle, B., Saranti, A., & Pfeifer, B. (2021). Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI. *Information Fusion, 71*, 28–37.

19. Aggarwal, C. C. (2023). *Neural networks and deep learning: A textbook* (2nd ed.). Springer.

20. Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Deep learning applications for COVID-19. *Journal of Big Data, 8*(1), Article 18.

21. Wei, J., Wang, X., Schuurmans, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems, 35*, 24824–24837.

22. Banks, A., & Porcello, E. (2020). *Learning React: Modern patterns for developing React apps* (2nd ed.). O'Reilly Media.

23. Subramanian, V. (2019). *Pro MERN Stack: Full stack web app development with Mongo, Express, React, and Node* (2nd ed.). Apress.

24. Achiam, J., Adler, S., Agarwal, S., et al. (2023). *GPT-4 technical report*. arXiv:2303.08774.

25. Ouyang, L., Wu, J., Jiang, X., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems, 35*, 27730–27744.