

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Розробка взаємодії системи та користувача на
прикладі застосунку психологічної підтримки»**

Виконав студент групи ЗПЗк-21-1
ЧЕРНЮК Антон Михайлович

Керівник роботи:
СЕМЕРІКОВ Сергій Олексійович

Консультант:
САВІЦЬКИЙ Роман Святославович

Рецензент:
ЧИЖМОТРЯ Олена Геннадіївна

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«26» лютого 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЧЕРНЮК Антон Михайлович**

Керівник роботи: **СЕМЕРІКОВ Сергій Олексійович**

Тема роботи: **«Розробка взаємодії системи та користувача на прикладі застосунку психологічної підтримки»**,

затверджена Наказом закладу вищої освіти від «23» лютого 2024 р. №75с

Вихідні дані для роботи: метою є розробка взаємодії системи та користувача на прикладі застосунку психологічної підтримки. Об'єктом дослідження є полегшення процесу листування терапевтів з пацієнтами та взаємодія системи з користувачем, пов'язана з користувацьким досвідом. Предметом дослідження є підходи та інструменти для реалізації користувацького інтерфейсу, та інструментів для операторів застосунку.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Семеріков С.О., Савіцький Р.С.	27.02.2024р.	27.02.2024р.
2	Семеріков С.О., Савіцький Р.С.	27.02.2024р.	27.02.2024р.
3	Семеріков С.О., Савіцький Р.С.	27.02.2024р.	27.02.2024р.

РЕФЕРАТ

Випускна кваліфікаційна робота складається з програмної реалізації взаємодії системи та користувача на прикладі застосунку психологічної підтримки, а також пояснювальної записки. Пояснювальна записка до дипломної роботи містить 85 сторінок, 32 ілюстрації, 19 таблиць, 1 додаток та 35 літературних джерел.

Метою роботи є дослідження та розробка взаємодії системи та користувача, яка надає користувачам можливість використання даного застосунку як електронного ресурсу для покращення психологічного стану та створення інструментів для полегшення взаємодії операторів з анонімними користувачами у чаті.

Основним інструментарієм при розробці є мова програмування JavaScript та фреймворки Next.js і Tailwind CSS. При розробці функціонального, сучасного, зручного і зрозумілого користувацького інтерфейсу було застосовано бібліотеки Radix UI, Shadcn UI та Lucide, що полегшили та пришвидшили розробку.

В даній роботі поставлено завдання для розробки та проаналізовано вимоги системи, оглянуто аналоги програмного забезпечення, що можуть скласти конкуренцію на ринку. Частиною записки також є опис процесу розробки бази даних та загальної структури застосунку з урахування потреб акторів. Наступним кроком викладено процес встановлення та налаштування застосунку, описано реалізацію користувацького інтерфейсу та приклади його тестування.

КЛЮЧОВІ СЛОВА: КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, ШТУЧНИЙ ІНТЕЛЕКТ, ВЗАЄМОДІЯ СИСТЕМИ ТА КОРИСТУВАЧА, ПСИХОЛОГІЧНИЙ СТАН, TAILWIND CSS, NEXTJS.

					ІПЗ.КР.Б – 121 – 24 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Чернюк А.М.			«Розробка взаємодії системи та користувача на прикладі застосунку психологічної підтримки» Пояснювальна записка	Літ.	Арк.	Аркушів
Керівник		Савіцький Р.С.					3	84
Рецензент		Чижмотря О.В.						
Зав. каф.		Вакалюк Т.А.				Житомирська політехніка, група ЗІПЗк-21-1		

ABSTRACT

The final qualification work consists of a software implementation of the interaction between the system and the user using the example of the application of psychological support, as well as an explanatory note. The explanatory note to the thesis contains 85 pages, 32 illustrations, 19 tables, 1 appendix and 35 literary sources.

The purpose of the work is to research and develop the interaction between the system and the user, which provides users with the ability to use this application as an electronic resource to improve the psychological state and create tools to facilitate the interaction of operators with anonymous users in the chat.

The main tools for development are JavaScript programming language and frameworks Next.js and Tailwind CSS. When developing a functional, modern, user-friendly and understandable user interface, Radix UI, Shadcn UI and Lucide UI libraries were used, which facilitated and accelerated the development.

This paper describes the analysis of the task for the development and development of system requirements, examined software analogues that can compete in the market. Part of the note is also a description of the process of developing the database and the overall structure of the application, taking into account the needs of the actors. The next step is the process of installing and configuring the application, the implementation of the user interface and examples of its testing are described.

KEYWORDS: USER INTERFACE, ARTIFICIAL INTELLIGENCE,
INTERACTION OF SYSTEM AND USER, PSYCHOLOGICAL STATE,
TAILWINDCSS, NEXTJS.

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	8
1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ	10
1.1 Постановка задачі.....	10
1.2 Аналіз аналогів програмного продукту	12
1.3 Вибір архітектури веборієнтованої системи психологічної підтримки.....	19
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	21
2. ПРОЄКТУВАННЯ ВЗАЄМОДІЇ СИСТЕМИ ТА КОРИСТУВАЧА НА ПРИКЛАДІ ЗАСТОСУНКУ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ.....	27
2.1 Опис основних акторів та загальної структури взаємодії системи та користувача	27
2.2. Розробка бази даних системи	32
2.3. Проєктування та реалізація алгоритмів роботи системи.....	42
2.4 Реалізація взаємодії системи та користувача	43
3. ІНТЕРФЕЙС ТА ПОРЯДОК ВЗАЄМОДІЇ З СИСТЕМОЮ ЗАСТОСУНКУ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ.....	53
3.1 Порядок встановлення та налаштування системи.....	53
3.2. Опис інтерфейсу програмного продукту	55
3.3. Тестування застосунку психологічної підтримки.....	65
ВИСНОВКИ	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70
ДОДАТКИ.....	73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AI – Artificial Intelligence (також ШІ) – галузь інформатики, що зосереджена на створенні розумних машин або систем, які здатні виконувати завдання, що зазвичай вимагають людського інтелекту.

API – Application Programming Interface – інтерфейс програмування застосунків, дозволяє різним програмам взаємодіяти між собою.

AWS – Amazon Web Services – набір облачних сервісів від Amazon, який надає віртуальні сервери, зберігання даних тощо.

CI/CD – Continuous Integration, Continuous Delivery – неперервна інтеграція і неперервне розгортання, практики в програмній інженерії, які забезпечують автоматизацію випуску і розгортання програмного забезпечення.

CSS – Cascading Style Sheets – мова каскадних таблиць стилів, використовується для оформлення вебсторінок.

CUID – Collision-resistant Unique Identifier – колізійно-стійкий унікальний ідентифікатор, використовується для створення унікальних ключів в базах даних.

DOM – Document Object Model – об'єктна модель документа, що дозволяє програмам змінювати структуру, стиль і зміст вебсторінок.

HTML – HyperText Markup Language – мова розмітки гіпертексту, використовується для створення структури вебсторінок.

ORM – Object Relational Mapper – технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних».

OTP – One-time Password – автоматично створений цифровий або буквено-цифровий рядок символів, що слугує для одноразової автентифікації користувача.

RPC – Remote Procedure Call – віддалений виклик процедур, технологія, яка дозволяє комп'ютерним програмам викликати процедури на іншому комп'ютері.

SQL – Structured Query Language – мова структурованих запитів, використовується для управління даними в реляційних базах даних.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

UI – User Interface – засіб зручної взаємодії користувача (людини) з інформаційною системою.

UX – User Experience – те, що людина відчуває при користуванні продуктом, системою чи сервісом (послугою). Основними об'єктами дослідження є враження, емоції та користь, отримані від взаємодії з продуктом.

СУБД – система управління базами даних – забезпечує зберігання, зміну та вибірку даних.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. У сучасному суспільстві питання психологічного здоров'я набуває все більшої актуальності. Війна, постійний тиск, висока інтенсивність життя, нестабільність, занурення у інформаційний потік та інші фактори суттєво впливають на психологічний стан людей. В цьому контексті важливим стає створення інструментів, які б допомагали підтримувати психологічне здоров'я та надавали можливість отримання психологічної підтримки в зручний для користувачів спосіб. Сучасні вебтехнології надають можливості для розробки застосунків, що могли б допомогти вирішити актуальну проблему. Оскільки проєкт передбачає реалізацію застосунку, що працював би на волонтерських засадах, дане рішення допомогло б підтримати найменш захищені верстви населення та допомогти їм подолати проблеми, що сприяло б подальшому розвитку.

Метою даної кваліфікаційної роботи є дослідження та розробка взаємодії системи та користувача на прикладі застосунку психологічної підтримки. Дана розробка передбачає реалізацію електронного ресурсу, який буде сприяти покращенню психологічного стану користувачів та інструментів для полегшення роботи волонтерів можливими засобами, такими як: використання штучного інтелекту, що проводив би аналіз діалогу з анонімним користувачем; можливість відправити посилання на дописи та вправи для покращення психологічного стану одразу у чат та створення зручних нотаток щодо почуттів підчас звернення.

Завданням даної кваліфікаційної роботи є:

- Проведення аналізу існуючих рішень у сфері застосунків психологічної підтримки.
- Продовженням є розробка архітектури застосунку, який би задовольняв потреби користувачів та операторів.
- Створення інтуїтивно зрозумілого та функціонального користувацького інтерфейсу за допомогою сучасних інструментів та бібліотек.
- Успішне проведення остаточного тестування застосунку з усуненням недоліків, отриманих в процесі.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єктом дослідження є процеси взаємодії користувачів із застосунками психологічної підтримки. Це включає вивчення того, як користувачі сприймають, використовують і взаємодіють із такими системами, а також виявлення факторів, що впливають на їх ефективність. Особлива увага приділяється аспектам, які визначають зручність використання, інтуїтивність інтерфейсу, а також загальну задоволеність користувачів від роботи з застосунком.

Предметом дослідження є розробка інтерфейсу користувача та реалізація функціоналу для забезпечення ефективної взаємодії між користувачами та системою в контексті застосунку психологічної підтримки. Це передбачає вивчення та впровадження практик дизайну інтерфейсів, що сприяють зручності використання, а також розробку технічних рішень, які забезпечують легкість у використанні системи операторами.

Розробка взаємодії системи та користувача згідно визначених вимог допоможе забезпечити якісну, доступну та зручну психологічну допомогу, що є надзвичайно важливим для підтримки психічного здоров'я населення.

Під час розробки взаємодії між системою та користувачем було проаналізовано реалізацію чату в реальному часі, використання штучного інтелекту в застосунку психологічної підтримки, а також здатність штучного інтелекту генерувати унікальну інформацію, бути об'єктивним і надійним. Це особливо важливо в контексті такої чутливої теми, як психологічна підтримка. Були опубліковані тези: ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В СИСТЕМАХ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ, XV Міжнародна науково-практична конференція «Комп'ютерні системи та мережні технології» [1], ІНТЕГРАЦІЯ МУЛЬТИПЛАТФОРМНИХ МЕСЕНДЖЕРІВ З ВИКОРИСТАННЯМ FASTAPI ТА WEBSOCKETS ДЛЯ СТВОРЕННЯ УНІФІКОВАНОЇ КОМУНІКАЦІЙНОЇ ПЛАТФОРМИ, XIV Міжнародна науково-технічна конференція «Інформаційно-комп'ютерні технології – 2024» [2].

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ

1.1 Постановка задачі

Дипломна робота на тему «Розробка взаємодії системи та користувача на прикладі застосунку психологічної підтримки» має на меті створення зручного та сучасного вебзастосунку для надання психологічної допомоги із застосуванням сучасних ІТ-технологій. Ціль проєкту – розробка гнучкої та ефективної системи управління застосунком, яка спростить моніторинг, управління та оптимізацію взаємодії між учасниками, а також контроль за контентом, пов'язаним з ментальним здоров'ям. Основні аспекти системи включають створення модулів для реєстрації користувачів, управління сесіями психологічної допомоги, аналізу поведінкових патернів за допомогою штучного інтелекту та автоматизацію процесів контенту.

Актуальність дослідження полягає у наростаючій потребі в психологічній підтримці населення, особливо в умовах стресу та психологічного навантаження, що спричинені сучасними соціокультурними та економічними реаліями. Застосунок цифрових технологій у цій сфері відкриває широкі можливості для забезпечення доступу до психологічної допомоги та покращення якості надання послуг.

Об'єктом дослідження є користувацька частина інтерфейсу та дизайн вебзастосунку, що є частинами взаємодії системи та користувача. Вони є ключовими аспектами у забезпеченні ефективної комунікації між системою та користувачем. Дані частини визначають зручність, доступність та естетичний аспект вебзастосунку, що має велике значення для його прийняття та використання широкою аудиторією. Дослідження цих аспектів дозволить виявити оптимальні рішення щодо побудови інтерфейсу та дизайну, спрямовані на підвищення задоволення користувачів та ефективність взаємодії з системою.

Предметом дослідження є методи та інструменти для розробки ефективної та сучасної взаємодії системи та користувача. Дослідження включатиме аналіз сучасних методів та інструментів веброзробки, таких як принципи UX/UI-

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

дизайну, адаптивний дизайн, а також техніки взаємодії з користувачем, такі як інтерактивність та персоналізація. Використання цих методів та інструментів сприятиме створенню вебзастосунку, який відповідатиме сучасним стандартам та вимогам користувачів, забезпечуючи оптимальний рівень зручності та ефективності взаємодії.

Аналіз передових практик у використанні штучного інтерфейсу дозволить ідентифікувати оптимальні рішення для поліпшення взаємодії між користувачем та системою. Адаптація цих практик до потреб конкретного застосунку психологічної підтримки забезпечить збалансований підхід до розробки, що враховує специфіку цільової аудиторії та контекст використання.

Оскільки головна сторінка вебзастосунку відіграє ключову роль у першому враженні користувача та його навігації в системі, їй приділялась особлива увага підчас проєктування користувацького інтерфейсу. Вона є точкою входу для більшості користувачів, тож важливо було забезпечити її зрозумілість, привабливість та функціональність, що підвищить ймовірність подальшого використання застосунку.

Крім технічної реалізації, проєкт передбачає аналіз потреб користувачів, що полягатиме в проведенні опитувань та аналізі зворотного зв'язку з метою зрозуміти їхні очікування, проблеми та потреби у психологічній підтримці. Це дозволить розробити взаємодію вебзастосунку, який відповідає реальним потребам користувачів та забезпечить найбільш зручну та ефективну взаємодію з системою.

Результатом роботи стало створення сучасної та зручної взаємодії вебзастосунку для надання психологічної допомоги, що ґрунтується на передових практиках у розробці інтерфейсу та взаємодії з користувачем. Взаємодія даного застосунку спрямована на поліпшення якості психологічної підтримки та забезпечення доступу до неї для більш широкої аудиторії, що має значний соціальний та психологічний вплив.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз аналогів програмного продукту

На сучасному ринку існує велика кількість систем психологічної підтримки, проте багато з них мають свої недоліки. Деякі з цих систем можуть бути складними у використанні або не забезпечувати достатньої конфіденційності для користувачів. Також існують системи, які є аналогами до розроблюваної, пропонуючи психологічну допомогу онлайн та через текстове листування. Однак, багато з них можуть бути недостатньо адаптованими до потреб користувачів або не забезпечувати достатньої якості обслуговування. Таким чином, розробка нової системи психологічної підтримки із застосуванням сучасних ІТ-технологій є актуальною та важливою задачею.

Для проведення порівняльного аналізу були вибрані виключно вітчизняні продукти, що пропонують онлайн психологічну підтримку, оскільки саме вони належать до категорійних рішень, що найбільш збігаються з метою та цілями розробки поточного проєкту.

Першим аналогом є платформа психотерапевтів hederu [3].

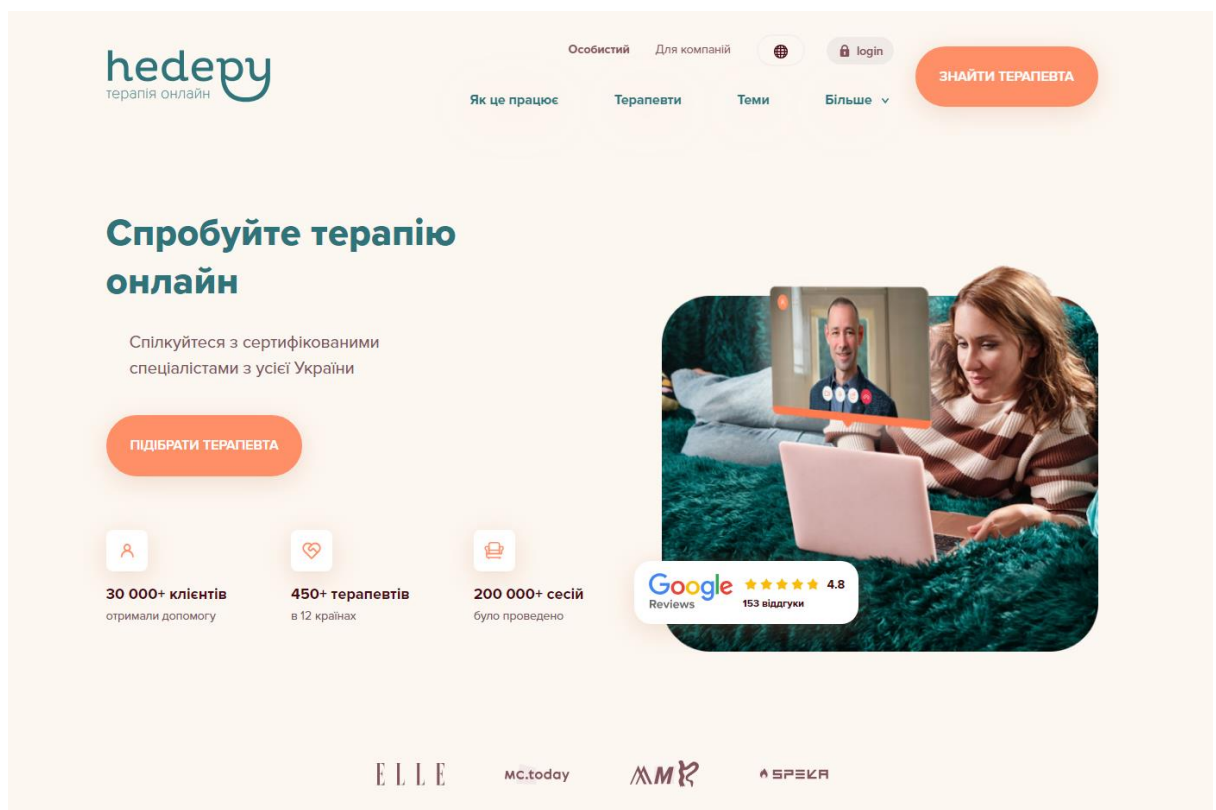


Рисунок 1.1. Головна сторінка сервісу.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Головна сторінка (рис 1.1) відзначається привабливим та ергономічним дизайном, де кольорова палітра та окремі елементи створюють гармонійне візуальне сприйняття. Інтуїтивно зрозуміле розташування основних елементів, таких як кнопки навігації та важливі функції, сприяє зручному взаємодії користувача з вебзастосунком. Приємний інтерфейс та ретельно продумана структура сторінки забезпечують надійний та ефективний спосіб отримання необхідної інформації.

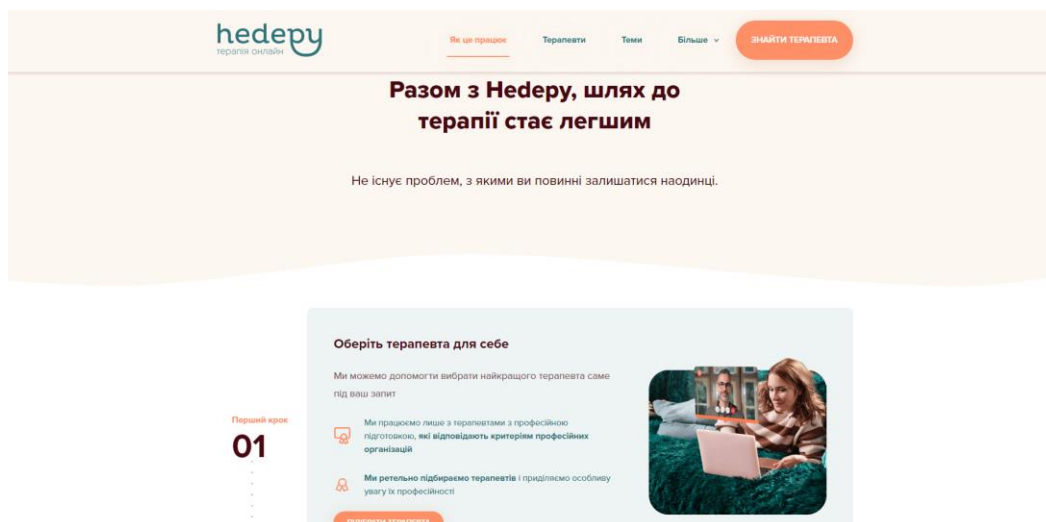


Рисунок 1.2. Сторінка – покрокова інструкція.

На сторінці «Як це працює» (рис. 1.2) користувачі знайдуть інструкції щодо запису на прийом до психотерапевта, включаючи процедуру вибору фахівця та часу сеансу. Описаний процес сеансу зазначає послідовність подій від зустрічі з психотерапевтом до завершення сесії та надання рекомендацій для подальшої роботи. Крім того, на сторінці представлені типові думки клієнтів перед першим сеансом, такі як тривога або невпевненість, а також відповіді на ці питання, що допомагають зняти страхи та уточнити очікування від терапевтичного процесу. Це створює прозору та дружню атмосферу для нових клієнтів та допомагає їм відчути комфорт під час першої зустрічі з психотерапевтом.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

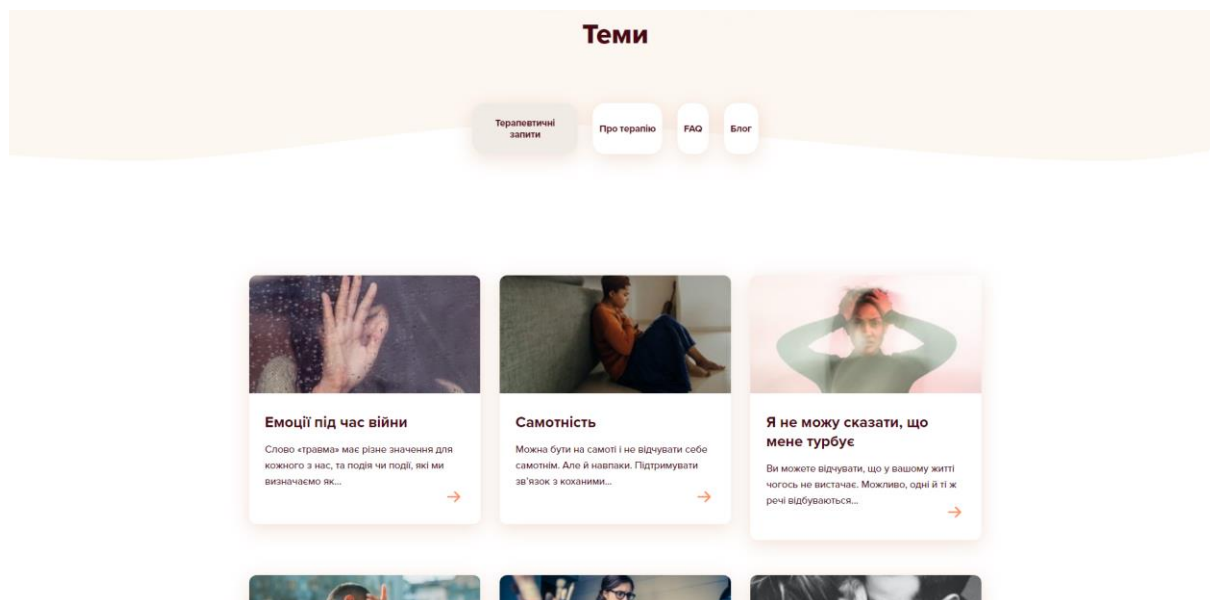


Рисунок 1.3. Сторінка «Тема».

Сторінка «Тема» (рис. 1.3) надає користувачам зручний доступ до різноманітної інформації про психотерапію через чотири вкладки. Вкладка «Терапевтичні запити» містить статті, які освітлюють різні теми, з якими звертаються до психотерапевтів, надаючи корисні поради та рекомендації, наприклад, «Емоції під час війни». «Про терапію» розділ містить базові відомості та відповіді на типові питання щодо психотерапії, прикладом є «Як підготуватись до першого сеансу». Вкладка «FAQ» надає користувачам швидку інформацію на технічні аспекти платформи та процесу психотерапії. Розділ «Блог» пропонує різноманітні статті та поради, які можуть бути корисними у різних життєвих ситуаціях та сприяти підтримці ментального здоров'я. Ця сторінка створена з метою надання користувачам не лише інформації, але й практичної підтримки та допомоги у вирішенні їхніх потреб.

Наступним аналогом програмного забезпечення є сервіс Tellme [4].

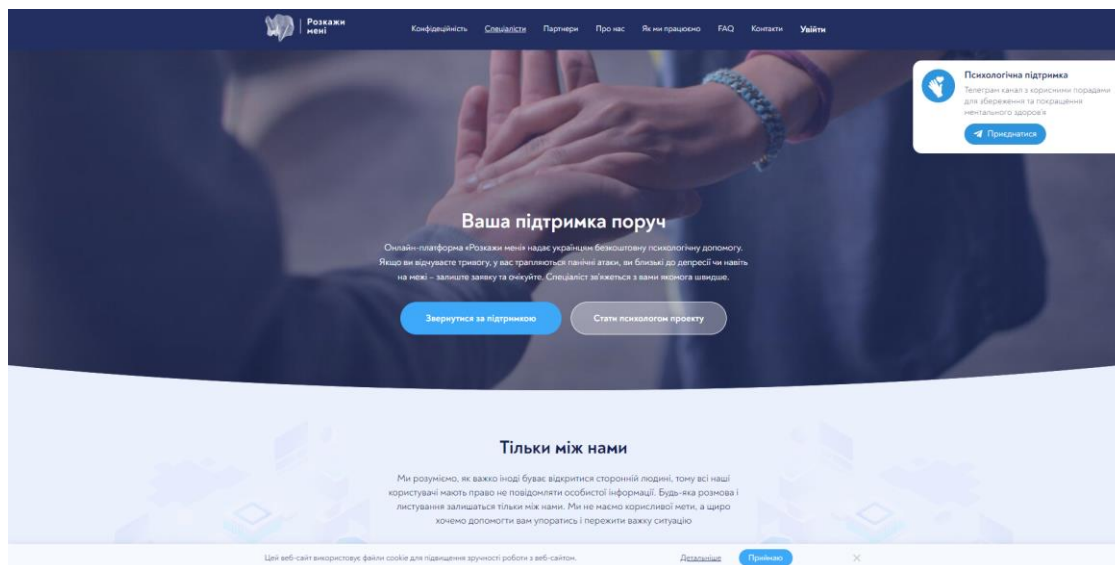


Рисунок 1.4. Головна сторінка сервісу.

Головна сторінка сайту (рис. 1.4) використовує холодну кольорову палітру, зосереджуючись на відтінках синього та сірого, що створює атмосферу спокою та стабільності. Незважаючи на те, що ці кольори можуть не зовсім відповідати тематиці психологічної підтримки, їхнє використання не є суттєвим недоліком, оскільки платформа пропонує безкоштовні послуги. Завдяки простому та інтуїтивно зрозумілому дизайну, користувачі можуть швидко знайти потрібну інформацію та скористатися психологічною підтримкою. Всі психологічні консультації проводяться через Інтернет, а всі фахівці, які беруть участь у проєкті, працюють на безоплатній основі.

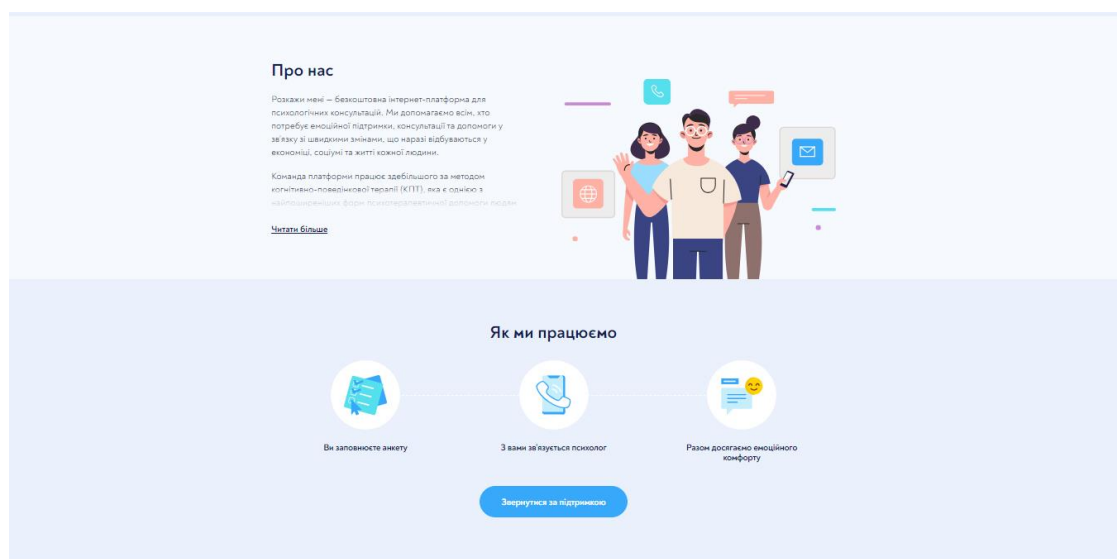


Рисунок 1.5. Інформація для користувачів.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

На даній платформі, на відміну від попередньої, відсутні окремі сторінки, але вся необхідна інформація (рис. 1.5) доступна на головній сторінці за посиланнями у верхньому колонтитулі. Це спрощує навігацію та забезпечує зручний доступ до всієї інформації для користувачів.

Крім вищезазначеного, решта вебсайту виконує функцію підтримки користувача та надання різноманітної інформації щодо принципів роботи платформи, процесу подання заявок, взаємодії з фахівцями та іншої корисної інформації.

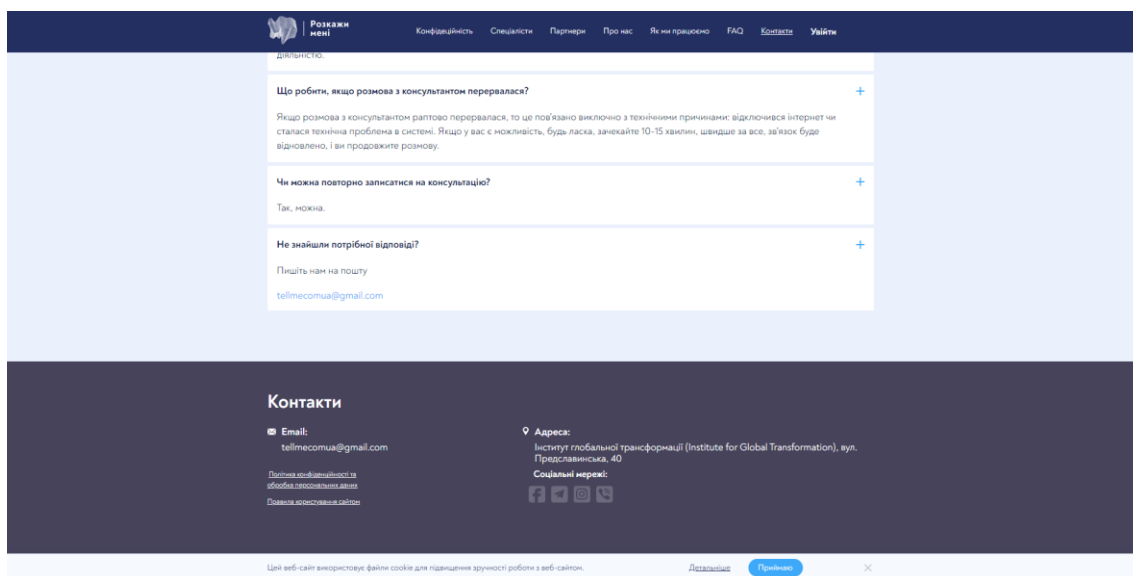


Рисунок 1.6. Частина сторінки із контактами та відповідями на запитання.

Система надає користувачам інформацію про доступ до різноманітних соціальних мереж, електронну пошту та фізичну адресу для зручності зв'язку (рис. 1.6). Проте вона не має можливості звертатися за підтримкою через текстовий чат прямо у системі.

Останній застосунок, що розглянуто як аналогічний – платформа психологічної підтримки Terlo [5].

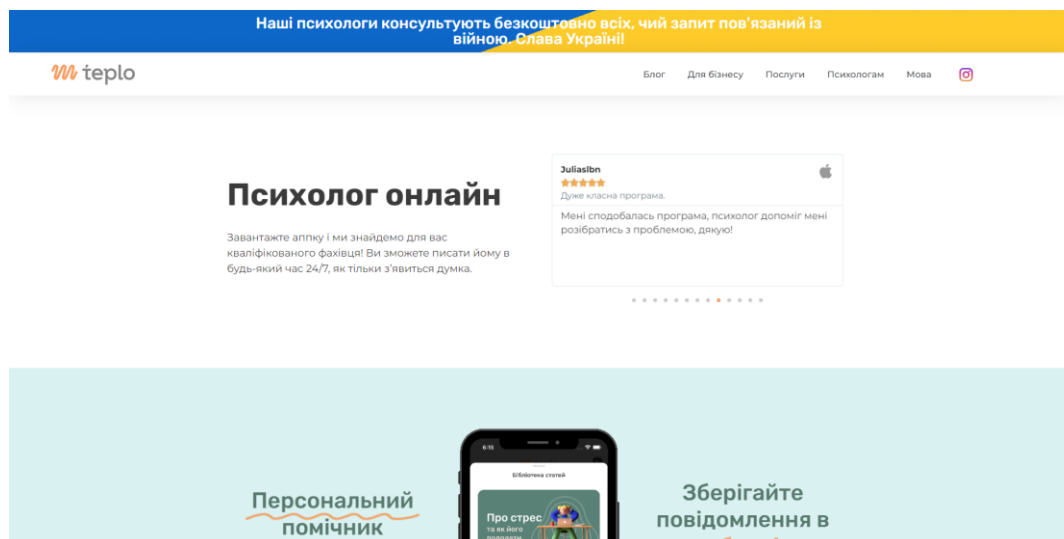


Рисунок 1.7. Головна сторінка сервісу.

Головна сторінка сайту (рис. 1.7) має досить приємну кольорову палітру, відповідну для галузі використання застосунку – психологічної підтримки. Білий та пастельні відтінки, такі як світло-блакитний, створюють атмосферу тепла та затишку. Незважаючи на те, що основні послуги платні та ціни високі, платформа пропонує безкоштовні консультації для постраждалих від війни осіб, що є важливим кроком у наданні доступної допомоги. Це додає гуманітарний аспект до вищезазначених послуг та робить платформу більш доступною для широкого кола користувачів.

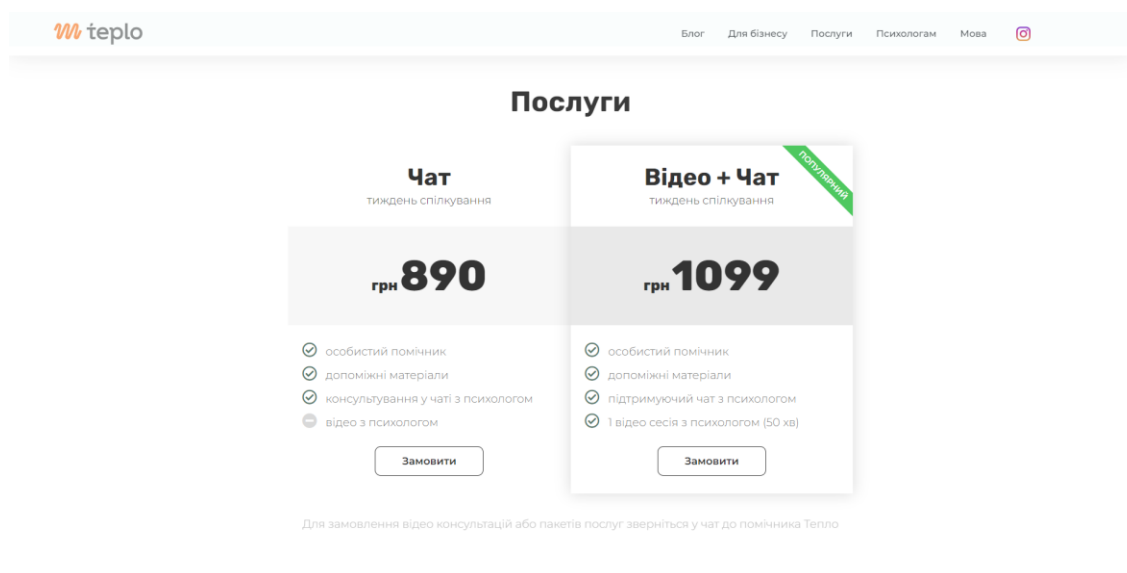


Рисунок 1.8. Монетизація сервісу.

Головною метою є монетизація застосунку, пропонуючи користувачам два типи послуг: з використанням відео-спілкування та без нього (рис. 1.8). Крім

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

того, платформа пропонує «пакети» відео-консультацій для різних потреб користувачів. Усі послуги включають допоміжні матеріали та «особистого помічника» на основі штучного інтелекту для кращого досвіду користувачів.

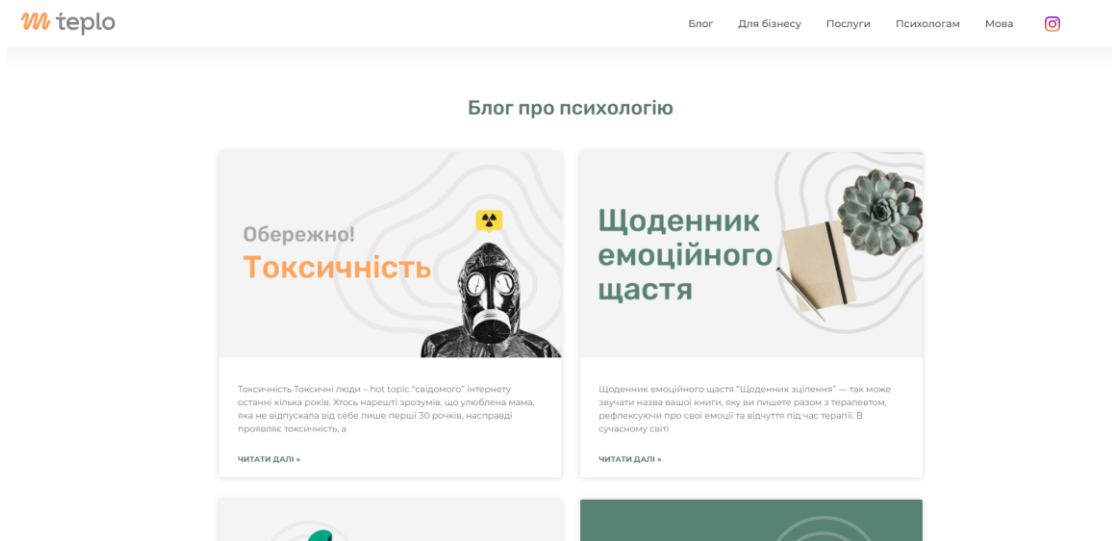


Рисунок 1.9. Сторінка «Блог».

На сторінці «Блог» (рис. 1.9) користувачі знайдуть цікаві та корисні статті-поради щодо підтримки психологічного здоров'я. Статті охоплюють різноманітні аспекти, включаючи стратегії зняття стресу, підвищення самоповаги та розвиток емоційної стійкості, проте ця можливість мало використовується та не є основною, оскільки основний контент доступний лише платним користувачам в межах застосунку. Крім того, платформа пропонує окрему сторінку, присвячену корпоративній психотерапії, яка може бути корисною для компаній, які прагнуть забезпечити підтримку своїм співробітникам у справах психічного здоров'я.

Вебсайт використовується переважно для маркетингових цілей, а весь бізнес-процес зосереджено в мобільному застосунку.

Таблиця 1.1.

Порівняння функціональних можливостей

Функція	Недєру	Розкажи мені	Терло	Розроблений застосунок
1	2	3	4	5
Матеріали для підтримки психологічного здоров'я	Так	Ні	Так	Так

Продовження таблиці 1.1.

1	2	3	4	5
Контакти	Так	Так	Ні	Ні
Відповіді на часті запитання	Так	Так	Ні	Так
Авторизація користувачів	Так	Ні	Так	Так
Можливість сплатити за послуги	Ні	Ні	Так	Так
Повний набір функцій при використанні стільникової версії	Так	Так	Ні	Так
Використання ІІІ	Ні	Ні	Ні	Так

Кожен з аналогів має свої особливості та переваги, однак найбільші досягнення з автоматизації процесів у сфері показав сервіс Тепло, оскільки пропонує користувачу найбільше цифрових рішень в мобільному застосунку, будучи єдиним сервісом, що має власний програмний продукт для мобільних пристроїв.

Отже, ключовими можливостями в новій системі психологічної підтримки повинні бути матеріали для підтримки психологічного здоров'я, чат з психологом, контакти пов'язаних установ, відповіді на часті запитання, авторизація адміністраторів, а за умови рішення монетизації системи також і авторизація користувачів. Система повинна мати повний функціонал при використанні стільникової версії, оскільки саме у ній найбільші можливості з надання цифрових рішень.

1.3 Вибір архітектури веборієнтованої системи психологічної підтримки

Те, що для розробки системи психологічної підтримки, що містить дописи та вправи, базу знань для операторів, чат та багато іншого, являється

надважливим рішенням. Воно підкріплене набором переваг, що є відповідними для проєкту такого типу.

Монолітна архітектура дає змогу контролювати типи та валідацію даних, для захисту інформації, що є особливо важливим у системі психологічної підтримки. Це гарантує цілісність та безпеку даних, а також сприяє спрощенню обробки даних на всіх рівнях програми, тож дозволяє сфокусуватись на логічній частині та зменшує ризик помилок, спрощуючи розробку та тестування.

Монолітна архітектура є досить надійною, причиною чого є інтеграція всіх компонентів системи в єдиному середовищі виконання. Оскільки всі компоненти програми доступні до обробки в одному місці, це полегшує виявлення та швидке усунення помилок.

Гнучкість монолітної архітектури дозволяє масштабуватись як вертикально (додаванням ресурсів до сервера), так і горизонтально (реплікацією інстанцій програми). Ця можливість дає змогу ефективно управляти навантаженням та оптимізувати продуктивність залежно від потреб.

Єдина кодова база монолітної архітектури значно спрощує процес розробки та підтримки системи. Саме через це розробники не повинні витрачати час на впровадження та тестування окремих компонентів, бо всі потрібні дані знаходяться в одному місці. Це уніфікує процес розгортання та полегшує розуміння системи. Використання єдиної кодової бази робить діагностику та виправлення помилок більш ефективними.

Всі компоненти монолітної архітектури використовують спільні ресурси, серед них бази даних та зовнішні служби. Це сприяє збереженню ресурсів, через те, що відпадає потреба дублювання даних та інфраструктури для мікросервісів, що є звичайним для мікросервісних архітектур.

Монолітні системи мають простішу структуру залежностей, порівняно з мікросервісними. Кожен компонент моноліту використовує спільний набір залежностей, що знижує ймовірність виникнення конфліктів та спрощує оновлення бібліотек, фреймворків.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

Монолітна архітектура дозволяє консолідувати всі заходи безпеки в одному місці, на відміну від мікросервісної, де вони розміщуються по багатьох мікросервісах. Це полегшує управління доступом, аутентифікацією, шифруванням даних та іншими важливими деталями безпеки, забезпечуючи захист всієї системи.

Монолітна архітектура пропонує достатньо переваг, таких як висока надійність, простота розробки та підтримки, ефективне використання ресурсів, просте управління залежностями та міцну систему безпеки. Ці речі роблять монолітну архітектуру привабливим вибором для розробки багатьох типів програмних систем.

Для зберігання даних в системі психологічної підтримки було обрано реляційну модель бази даних. Хоча реляційні бази даних інколи потребують більший обсяг ресурсів та мають обмежену масштабованість порівняно з деякими нереляційними альтернативами, дані недоліки компенсуються можливістю оптимізувати обробку запитів. Завдяки оптимізації, кількість запитів до бази даних значно знижується, що мінімізує потенційні обмеження, пов'язані з вибором реляційної системи. В результаті, переваги реляційної моделі у встановленні складних зв'язків перевищують згадані обмеження, роблячи її виправданим вибором для потреб даної системи.

Комбінація монолітної архітектури та проксі-сервера для системи психологічної підтримки не лише виправдана з технічної точки зору, але й стратегічно вигідна з погляду довгострокового розвитку та адаптації до майбутніх викликів. Такий підхід дозволяє поєднати переваги монолітної архітектури – надійність та простоту управління – з перевагами проксі-сервера – гнучкістю та високою пропускнуною спроможністю. Це відкриває нові можливості для створення масштабованих, безпечних та високопродуктивних вебсистем.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Для розробки вебзастосунку «система психологічної підтримки» було обрано JavaScript [6] як основну мову програмування. Це рішення було прийнято

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

з кількох ключових причин. По-перше, JavaScript є однією з найбільш поширених мов програмування для веброзробки, що забезпечує підтримку JavaScript на усіх нині існуючих веб браузерях, а також, доступ до великої кількості бібліотек і фреймворків. Це дозволяє значно пришвидшити процес розробки та знижує витрати на підтримку проєкту. По-друге, JavaScript має широку спільноту розробників, що сприяє розвитку нових технологій на базі даної мови програмування, а також їх поширення та, відповідно, можливості їх вивчення. Кількість та якість документації, навчальних матеріалів та форумів підтримки є важливим фактором для забезпечення стабільної та ефективної розробки [7].

Логічним вибором бібліотеки для розробки фронтенду, що включає в себе всі вище перелічені переваги, є React [8]. Вона пропонує високий рівень абстракції компонентів, що спрощує процес розробки та підтримки вебзастосунків. Однією з ключових особливостей React є використання віртуального DOM. Віртуальний DOM – це представлення реального DOM, яке React використовує для ефективного оновлення інтерфейсу користувача [9]. Завдяки його використанню React може мінімізувати кількість операцій з реальним DOM, що значно підвищує продуктивність вебзастосунку.

Враховуючи вибір JavaScript та React, для проєкту було вирішено використовувати стек веброзробки T3 [10]. Стек T3 включає в себе сучасні інструменти, які забезпечують високу продуктивність та ефективність розробки. Серед основних переваг можна виділити високу продуктивність, гнучкість та зручність використання. Стек T3 дозволяє створювати масштабовані та надійні вебзастосунки з багатим функціоналом, що є ключовим для системи психологічної підтримки [11].

Основою стеку T3 є Next.js, tRPC, Tailwind, TypeScript, Prisma та NextAuth, кожен з яких надає унікальні переваги та разом створюють надійну інфраструктуру для проєкту.

Next.js [12], фреймворк для React, забезпечує ефективний серверний рендеринг і генерацію статичних сайтів, що сприяє високій продуктивності та

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

швидкому завантаженню сторінок. Це є критично важливим для системи психологічної підтримки, де швидкий доступ до ресурсів та миттєвий відгук інтерфейсу підвищують користувацький досвід. Крім того, Next.js спрощує організацію маршрутизації та покращує SEO оптимізацію, що сприяє кращому залученню користувачів і пошуковій видимості, допомагаючи системі бути доступною для більшої кількості людей, які шукають психологічну підтримку онлайн [13].

Tailwind CSS [14], в свою чергу, пропонує практичний підхід до створення стилів, що значно знижує ресурси, необхідні для розробки інтерфейсу користувача. Це дозволяє швидко створювати красиві та функціональні інтерфейси без необхідності написання великої кількості кастомних CSS [15] класів, використовуючи стилі безпосередньо у тегах, де вони необхідні, також, за необхідності зручно створити глобальні стилі, які можна застосовувати до усіх компонентів проєкту. Також, даний фреймворк пропонує заздалегідь створені кольорові гами та спрощує адаптивний дизайн за допомогою вбудованих класів. У контексті системи психологічної підтримки, де дизайн має бути ненав'язливим, інтуїтивно зрозумілим і привабливим, Tailwind, полегшує задачу створення єдиного стилю для всіх компонентів застосунку, що сприяє кращій користувацькій взаємодії.

TypeScript [16], як надбудова над JavaScript, додає статичну типізацію, що допомагає уникнути багатьох помилок під час розробки та є гарним вибором для застосування з бібліотекою Zod. TypeScript допомагає розробникам писати більш безпомилковий код, вказуючи на місця їх можливого виникнення, що знижує ризики. Типізація також полегшує інтеграцію з іншими інструментами та бібліотеками, що робить розробку більш ефективною та менш схильною до помилок [17].

Prisma [18], як ORM інструмент, забезпечує зручну та ефективну роботу з базою даних. В контексті системи психологічної підтримки, де обробка та зберігання даних є критично важливими, Prisma надає API з типізацією для доступу до даних, що спрощує процес написання запитів та забезпечує високу

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

продуктивність. Це дозволяє розробникам зосередитися на логіці застосунку, а не на складнощах роботи з базою даних, що підвищує ефективність розробки та забезпечує надійність збереження даних користувачів [19].

Нарешті, NextAuth [20] є ключовим елементом для забезпечення безпеки та управління доступом у системі психологічної підтримки. Ця бібліотека для аутентифікації та авторизації користувачів у Next.js застосунках надає зручний та безпечний спосіб керування доступом до різних частин застосунку. Використання NextAuth дозволяє легко інтегрувати різні методи аутентифікації, включаючи OAuth та одноразові паролі (OTP), що забезпечує високу безпеку користувацьких даних та підвищує довіру користувачів до системи [21].

Об'єднуючи всі ці інструменти, стек T3 створює потужну та ефективну платформу для розробки системи психологічної підтримки. Це забезпечує високу продуктивність, безпеку, зручність використання та масштабованість, що є критично важливими для успішної реалізації та подальшої підтримки проєкту.

Для генерації та використання API у даному проєкті було обрано tRPC [22] – фреймворк RPC (віддаленого виклику процедур), що спрощує розробку API. Він не вимагає окремих визначень схем, пропонуючи низку переваг у порівнянні з традиційними методами RPC. Однією з його функцій є забезпечення плавної взаємодії між різними частинами системи. Завдяки безпеці типів, tRPC спрощує процес розробки, зменшуючи кількість помилок та прискорюючи його. Це робить використання застосунку більш безпечним, а автоматизація процесів зменшує час, витрачений на розробку продукту. Не можна не зазначити і легку інтеграцією фреймворку з Next.js [23].

Для уникнення помилок під час виконання фінального продукту та встановлення жорсткої типізації даних було обрано бібліотеку Zod [24]. Ось чому Zod вигідно відрізняється від конкурентів:

- Інтеграція з TypeScript: Zod дозволяє використовувати переваги статичної типізації TypeScript, роблячи код більш надійним та стійким до помилок.
- Широке використання: Завдяки великій та активній спільноті розробників, Zod гарантує стабільність бібліотеки та доступність якісної документації.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Для розробки системи психологічної підтримки також було вирішено використовувати бібліотеку Radix UI [25] та компоненти Shadcn [26]. Ці інструменти додають значну цінність проєкту, забезпечуючи як естетичну привабливість, так і функціональність інтерфейсу користувача.

Radix UI є набором низькорівневих, доступних компонентів для створення якісних користувацьких інтерфейсів. Основною перевагою Radix UI є його фокус на доступність та інтерфейсні стандарти. Компоненти Radix UI розроблені з урахуванням потреб користувачів з обмеженими можливостями, що забезпечує інклюзивність застосунку. Для даного проєкту це особливо важливо, оскільки він орієнтований на широку аудиторію, включаючи людей з різними рівнями цифрової грамотності та фізичними обмеженнями. Використання Radix UI допомагає створювати інтерфейси, які є доступними для всіх користувачів, що підвищує загальну якість та ефективність застосунку [27].

Компоненти Shadcn доповнюють Radix UI, забезпечуючи готові до використання високорівневі UI-компоненти. Основною перевагою Shadcn є його інтеграція з сучасними фреймворками та бібліотеками, такими як Tailwind CSS, що забезпечує узгодженість стилів та легкість у використанні. Проте, як і у випадку з Radix UI, використання Shadcn може вимагати певних зусиль для налаштування та інтеграції з іншими частинами стеку. Незважаючи на це, переваги швидкого розвитку та високої якості інтерфейсу переважають ці потенційні недоліки.

Використання Radix UI та Shadcn у розробці користувацької частини системи психологічної підтримки обґрунтоване їхніми перевагами, які значно підвищують якість та ефективність застосунку. Radix UI забезпечує високу доступність та гнучкість інтерфейсів, що є критично важливим для широкої аудиторії користувачів. Shadcn доповнює Radix UI, надаючи готові до використання компоненти, які можна змінити під свої потреби. Разом ці інструменти створюють міцну основу для побудови якісного та функціонального вебзастосунку, що відповідає вимогам сучасних стандартів веброботи [28].

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

Система використовує сервіси Amazon для надсилання електронних повідомлень. Для їх реалізації було задіяно набір модулів з бібліотеки AWS SDK v3 [29], а також консольний застосунок AWS CLI v2 [30].

Для написання вебзастосунку «система психологічної підтримки» було обрано Visual Studio Code (VSCode) [31] як основне середовище розробки. VSCode пропонує зручний та інтуїтивно зрозумілий інтерфейс, що значно покращує ефективність роботи розробників. Завдяки широкому спектру розширень, таких як підтримка TypeScript, інтеграція з Git, автозавершення коду та дебагінг, VSCode забезпечує комфортну та продуктивну роботу над проєктом. Вбудований термінал та можливість налаштування робочого середовища дозволяють швидко виконувати завдання без необхідності переходу між різними інструментами. Крім того, велика спільнота користувачів та багата база ресурсів роблять VSCode ідеальним вибором для розробки складних вебзастосунків.

Для створення вебзастосунку обрано наступні інструменти:

1. Основна мова програмування: Typescript.
2. Бібліотека для валідації, санітизації та типізації: Zod.
3. Бібліотека для розробки фронтенду: React.
4. Бібліотека React-компонентів для пришвидшення розробки фронетнду: Radix UI, Shadcn UI, React Charts.
5. Мета-фреймворк для спрощення та пришвидшення фулстек-розробки: Nextjs.
6. Бібліотека-ОРВ для спрощення взаємодії із СУБД: Prisma.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

2. ПРОЄКТУВАННЯ ВЗАЄМОДІЇ СИСТЕМИ ТА КОРИСТУВАЧА НА ПРИКЛАДІ ЗАСТОСУНКУ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ

2.1 Опис основних акторів та загальної структури взаємодії системи та користувача

Важливою частиною розробки є проєктування. Для початку необхідно проаналізувати групи користувачів та їх дії, та описати загальну структуру вебзастосунку з варіантами використання.

Головними акторами даного проєкту є анонімні користувачі, оператори та користувачі, авторизовані для виконання певних дій, що представлені у таблиці 2.1.

Таблиця 2.1

Користувачі

Актор	Опис
Анонімний користувач	Має можливість перегляду головної сторінки, доступ до перегляду сторінок з дописами та вправами, сторінки з питаннями та відповідями, має посилання на чати, куди може звернутися за психологічною підтримкою та може підтримати проєкт фінансово за допомогою сторінки «Підтримати»
Оператор	Має права на усе те ж, що може робити анонімний користувач, також додатково має доступ до чатів та бази знань.
Авторизований користувач	Має ті ж права, що й анонімний користувач, та надані іншим авторизованим користувачем дозволи.

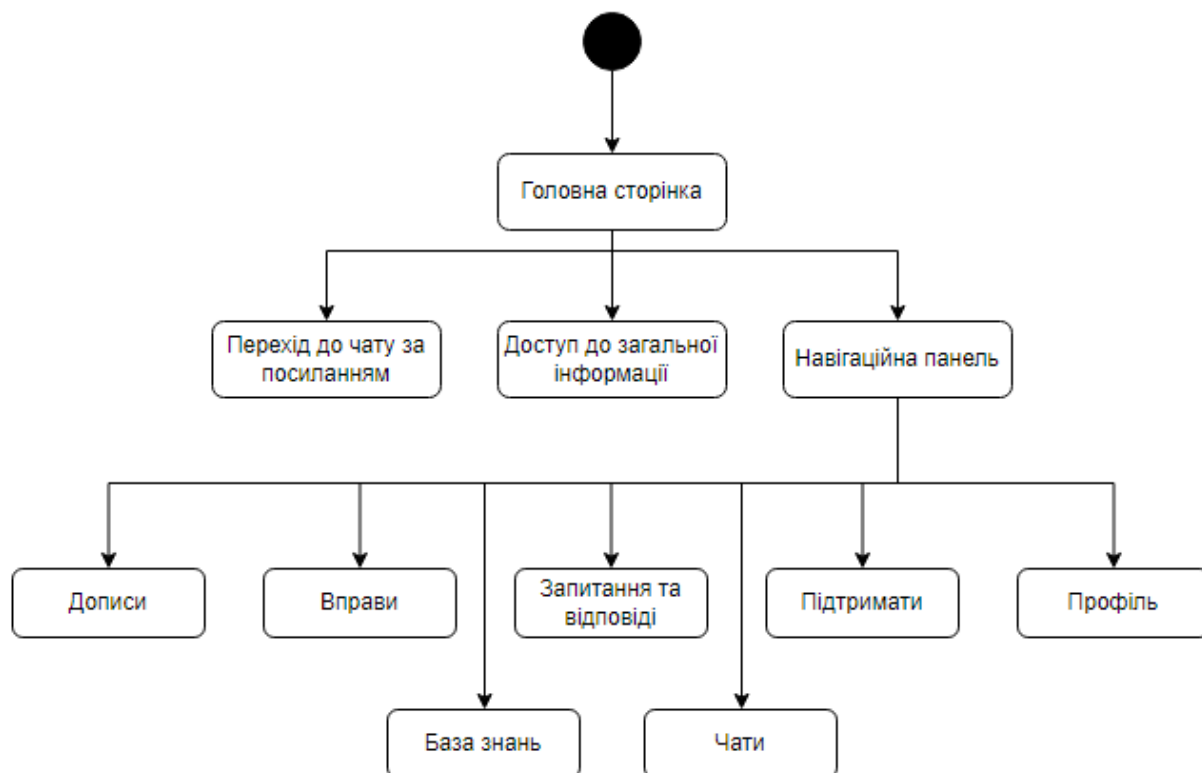


Рисунок 2.1 – Загальна структура вебзастосунку для користувачів

Основні компоненти вебзастосунку представлені на рисунку 2.1 у вигляді блок-схеми. При переході на вебзастосунок користувач потрапляє на головну сторінку, на якій містяться посилання для переходу до чату, доступ до загальної інформації та навігаційна панель. Навігаційна панель надає користувачам доступ до усіх сторінок та функцій застосунку. На ній будуть розміщені посилання на загальнодоступні сторінки, сторінки для операторів та кнопка функцій профілю. Також у навігаційній панелі повинне бути посилання на профіль користувача. Структура застосунку надає можливість зрозумілого доступу до функціоналу, а навігаційна панель забезпечує надання доступу до спеціалізованих функцій різних інформаційних розділів.

Короткий опис кожної з основних сторінок, що містяться у навігаційній панелі описано у таблицях 2.2 – 2.9:

Таблиця 2.2

Опис сторінки «Дописи»

Основні актори	Анонімний користувач, оператор, авторизований користувач
Короткий опис	Користувачі мають можливість переглянути список дописів, скористатись пошуком щоб перейти на певний допис та відкрити сторінку допису. Авторизовані користувачі мають функцію їх видалення.

Таблиця 2.3

Опис сторінки «Вправи»

Основні актори	Анонімний користувач, оператор, авторизований користувач
Короткий опис	Користувачі мають можливість переглянути список вправ, скористатись пошуком щоб перейти на певну з них та відкрити сторінку з вправою для покращення ментального здоров'я. Видаляти вправи можуть користувачі з наданими на це правами.

Таблиця 2.4

Опис сторінки «База знань»

Основні актори	Оператор, авторизований користувач
Короткий опис	Сторінка «База знань» є сховищем інформаційних ресурсів, включаючи статті, наукові дослідження, довідкові матеріали та інші корисні документи, які допомагають операторам у роботі з чатом та анонімними користувачами. Авторизовані користувачі з відповідними правами, в свою чергу, відповідають за створення, та редагування їх вмісту.

Таблиця 2.5

Опис сторінки «Запитання та відповіді»

Основні актори	Анонімний користувач
Короткий опис	Користувачі мають доступ до перегляду відповідей на запитання, що можуть містити корисну інформацію. Також є можливість залишити власне запитання, вказавши своє ім'я, електронну пошту для того, щоб на нього могли відповісти. Надалі це питання та відповідь на нього може бути опубліковане авторизованими користувачами на адміністративній сторінці.

Таблиця 2.6

Опис сторінки «Чати»

Основні актори	Оператор, авторизований користувач
Короткий опис	Оператори, коли в мережі, можуть отримувати чати, що розподіляються автоматично з урахуванням як суб'єктивного відчуття зайнятості, так і спеціальних метрик. Авторизованим користувачам з відповідною кваліфікацією також може бути наданий доступ до сторінки з чатами.

Таблиця 2.7

Опис сторінки «Підтримати»

Основні актори	Анонімний користувач
Короткий опис	Сторінка є джерелом фінансової підтримки проєкту. На ній описано заклик до підтримки та реалізовано можливість фінансової допомоги. Користувач може обрати тип допомоги, ввести певну суму та валюту та перейти на сторінку оплати, після якої буде переспрямований на сторінку подяки.

Таблиця 2.8

Опис сторінки «Профіль»

Основні актори	Оператор, авторизований користувач
Короткий опис	Профіль розроблений для операторів та авторизованих користувачів. На цій сторінці вони мають змогу редагувати свої дані. Перед переходом на сторінку користувач повинен увійти в обліковий запис.

Таблиця 2.9

Опис сторінки «Вхід»

Основні актори	Оператор, авторизований користувач
Короткий опис	Сторінка, розроблена для авторизації операторів та користувачів з додатковими правами (адміністраторів, модераторів).

Дані сторінки (таблиці 2.2 – 2.9) дозволяють задовольнити потреби як анонімного користувача, так і оператора або авторизованих користувачів. Кожна сторінка має чітко визначене призначення і сприяє ефективній підтримці та покращенню психологічного здоров'я користувачів.

2.2. Розробка бази даних системи

Для управління базами даних була обрана система PostgreSQL, оскільки при початковому аналізі проєкту було виявлено високу необхідність у зв'язках та інших функціях, які надають лише реляційні бази даних. Обрана СУБД є стандартом для широкого спектру вебпроєктів [32].

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

Діаграма на рисунку 2.2 показує всі таблиці в базі даних і зв'язки між ними. Її було згенеровано за допомогою PgAdmin, який є частиною інструментарію екосистеми PostgreSQL [33].

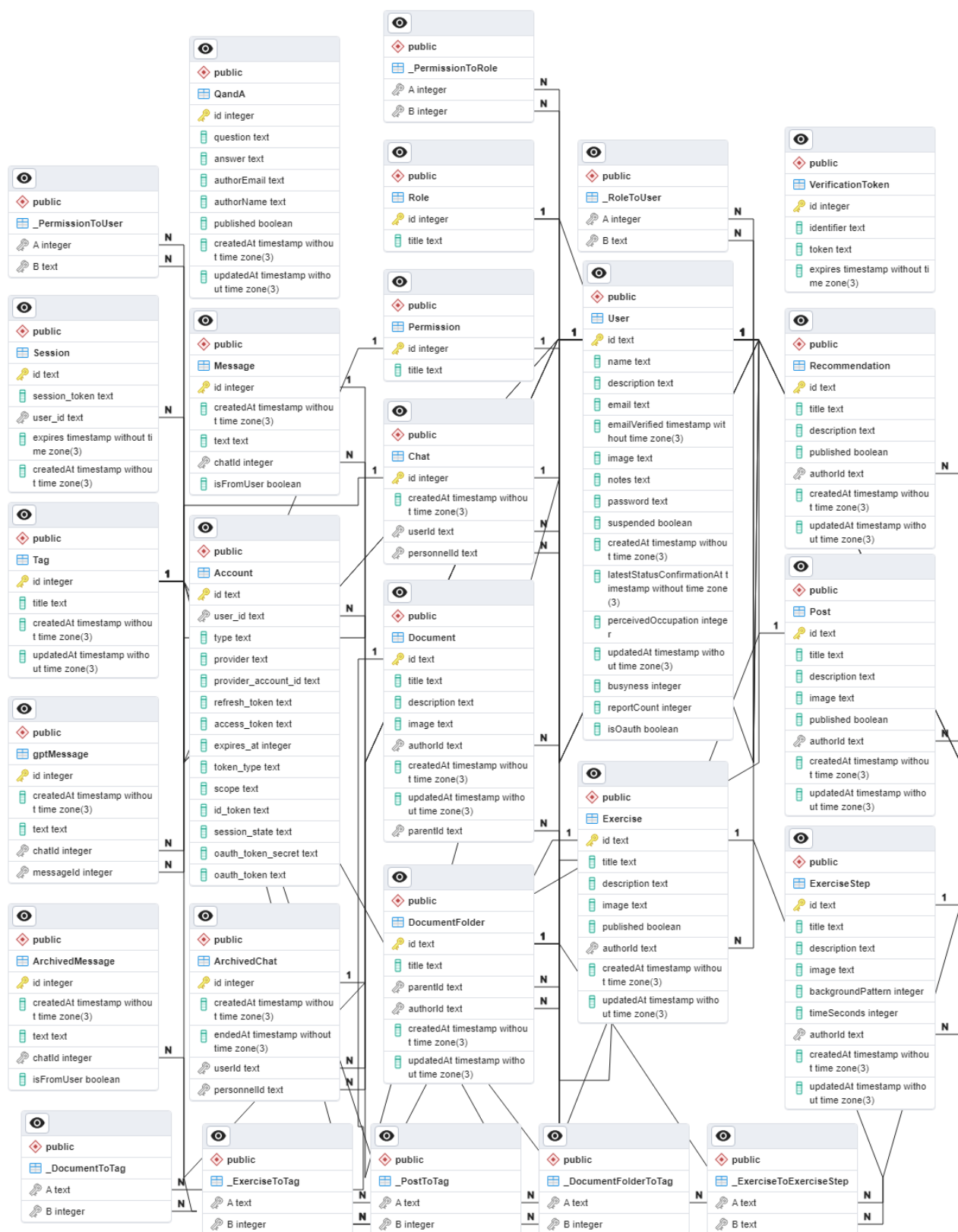


Рисунок 2.2. Діаграма «сутність-зв'язок» для бази даних.

У таблиці 2.10 відображено та описано кожну Prisma-модель, що також відповідають таблицям у базі даних, зображеним на рисунку 2.2.

Таблиця 2.10

Опис моделей Prisma, що відповідають таблицям в базі даних

Модель	Опис
1	2
User	Загальна модель користувача, яка включає базові дані (ім'я, опис, електронну адресу), статус онлайн, ролі, сесії, облікові записи та зв'язки з чатами та контентом.
VerificationToken	Використовується для зберігання токенів верифікації, що містять ідентифікатор, унікальний токен і дату закінчення.
Account	Модель для зберігання інформації про облікові записи соціальних мереж або інших зовнішніх провайдерів. Включає токени, типи акаунтів та ідентифікатори.
Session	Зберігає дані про сесії користувачів, такі як токен сесії, ідентифікатор користувача та термін дії сесії.
Permission	Визначає дозволи, які можуть бути призначені користувачам або ролям.
Chat	Відображає чати між користувачами та персоналом, зберігаючи пов'язані повідомлення та дати створення.
Message	Модель для зберігання тексту повідомлень у чаті, включаючи інформацію про те, чи було повідомлення від користувача.

1	2
ArchivedChat	Зберігає інформацію про архівовані чати, включаючи час завершення та пов'язані повідомлення.
ArchivedMessage	Модель для зберігання архівованих повідомлень, що належать до архівованих чатів.
Recommendation	Модель рекомендацій, що може містити заголовки, описи та зображення.
Post	Публікації користувачів, які можуть містити опис, зображення та теги для категоризації.
Exercise	Відображає вправи, які включають назву, опис, теги та кроки вправ.
ExerciseStep	Описує окремі кроки в рамках вправи, включаючи заголовок, опис та зображення.
Document	Модель для документів, які можуть бути частиною папок документів і містити теги для категоризації.
DocumentFolder	Папки для організації документів, можуть містити батьківські папки та дочірні папки.
QandA	Зберігає запитання та відповіді для розділу Q&A, включаючи інформацію про автора.
gptMessage	Відповідає за облік звернень до OpenAI API. Використовується для відображення попередніх звернень та заборони повторних.

Надалі сутності бази даних, описані у таблиці 2.10 та створені для реалізації взаємодії системи та користувача на прикладі системи психологічної

підтримки, буде розглянуто більш детально, включаючи поля та функціонал, який вони забезпечують.

Таблиця 2.11

Допоміжна таблиця в базі даних

Таблиця	Опис
_ExerciseToExerciseStep	Пов’язує вправи та їх кроки

У таблиці 2.11 описана допоміжна таблиця, яку створює Prisma для зв’язків типу багато-до-багатьох. Ці таблиці не мають власних моделей, оскільки створюються «неявним» оголошенням у моделях, які вони пов’язують.

Таблиця 2.12

Опис стовпців таблиці gptMessage

Назва	Опис
id	Унікальний ідентифікатор повідомлення.
createdAt	Час і дата створення повідомлення.
text	Текст повідомлення.
chatId	Ідентифікатор чату, до якого належить повідомлення.
messageId	Ідентифікатор повідомлення, що використовується для відстеження в межах чату.

Модель gptMessage (таблиця 2.12) в Prisma забезпечує облік звернень до OpenAI API [34]. Ця модель дозволяє відображати історію запитів і забороняти повторні звернення. Ідентифікатор кожного запису генерується автоматично і зберігається в полі id. Поле createdAt визначає час створення запису та

автоматично встановлюється в поточний момент створення. Звернення містить текст запиту в полі text. Кожне звернення асоціюється з конкретним чатом через поле chat, яке використовує зовнішній ключ chatId і забезпечує каскадне видалення. Опціонально, звернення може мати зв'язок з іншим повідомленням через поле message, яке використовує зовнішній ключ messageId та дозволяє встановити належність до попереднього повідомлення, з можливістю його видалення без видалення повідомлення.

Таблиця 2.13

Опис стовпців таблиці Post

Назва	Опис
id	CUID поста.
title	Заголовок поста.
description	Опис поста.
image	Шлях до зображення поста.
published	Чи опубліковано пост.
authorId	ID автора поста.
createdAt	Дата та час створення поста.
updatedAt	Дата та час оновлення поста.

Модель Post (табл. 2.13) описує публікації, які містять заголовок, опис та опціонально зображення. Публікації зазвичай пов'язані з автором. Поле published дозволяє контролювати видимість публікації на випадок відкладеної публікації чи при виявленні неточної інформації у публікації. Час створення та останнього оновлення також відстежуються.

Опис стовпців таблиці Recommendation

Назва	Опис
id	Унікальний ідентифікатор рекомендації.
title	Назва рекомендації.
description	Детальний опис рекомендації.
published	Чи опублікована рекомендація.
authorId	Ідентифікатор автора рекомендації.
createdAt	Дата і час створення запису.
updatedAt	Дата і час останнього оновлення запису.

Модель Recommendation (табл. 2.14) визначає рекомендації, що відображаються пліч-о-пліч з дописами. Кожна рекомендація має унікальний ідентифікатор, назву, опис, і може бути пов'язана з автором через зв'язок з таблицею користувачів. Властивість published вказує, чи є рекомендація доступною для публіки. Дати створення та останнього оновлення відстежуються автоматично, що дозволяє забезпечити актуальність інформації. Рекомендації відображаються користувачам випадковим чином.

Опис стовпців таблиці Exercise

Назва	Опис
id	Ціле число, що унікально ідентифікує кожне вправу в базі даних.
title	Назва вправи.
description	Детальний опис вправи.
image	Зображення, що ілюструє вправу.
published	Булевий індикатор, який показує, чи опублікована вправа.
authorId	Ідентифікатор автора вправи.
createdAt	Дата і час створення запису.
updatedAt	Дата і час останнього оновлення запису.

Модель Exercise (табл. 2.15) призначена для управління вправами на сайті. Кожна вправа включає заголовок, опис та можливо зображення. Вправи можуть бути теговані та містити кроки, що представлені моделлю ExerciseStep. Аналогічно іншим моделям, вправа має поля для відстеження авторства, публікації та часових відміток створення та оновлення.

Опис стовпців таблиці ExerciseStep

Назва	Опис
id	ідентифікує кожен крок у вправі.
title	Назва кроку вправи.
description	Детальний опис кроку вправи.
image	Зображення, що ілюструє крок.
timeSeconds	Час у секундах, необхідний для виконання кроку.
authorId	ID автора, який створив крок.
createdAt	Час створення запису кроку вправи.
updatedAt	Час останнього оновлення запису кроку вправи.

Модель ExerciseStep (табл. 2.16) використовується для деталізації кроків у вправах. Кожен крок містить заголовок, опис, опціонально зображення, і може містити тривалість виконання у секундах, що дозволяє користувачу зберігати ритм чи навпаки затримувати його на кроках як-от глибоке, спокійне дихання. Кроки можуть бути пов'язані з конкретними вправами, а також містять інформацію про автора та час створення.

Опис стовпців таблиці QandA

Назва	Опис
id	Ціле число, що унікально ідентифікує кожне запитання та відповідь в базі даних.
question	Текст запитання, поставленого користувачем.
answer	Текст відповіді на запитання.
authorEmail	Електронна пошта автора, який створив запис.
authorName	Ім'я автора, який створив запис.
published	Логічне значення, що показує, чи опубліковано запис.
createdAt	Часова мітка, що позначає коли запис було створено.
updatedAt	Часова мітка, що позначає коли запис було оновлено.

Модель QandA (табл. 2.17) використовується для зберігання питання та слідує за ним відповіді від психологів та адміністрації сайту психологічної підтримки. Кожен запис містить унікальний ідентифікатор id, який автоматично збільшується з кожним новим записом. Поле question зберігає текст питання, а answer — текст відповіді, який може бути незаповненим, поки відповідь не буде надана. Додатково зберігаються authorEmail та authorName для інформування користувача щодо відповіді на його запитання за наданою адресою електронної пошти. Поле published визначає, чи є питання публічно доступним. Час

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

створення і останнього оновлення запису фіксується в полях createdAt та updatedAt відповідно.

Результатом проєктування стала база даних, що відповідає встановленим вимогам до системи та потребам спроектованої системи адміністрування застосунку психологічної підтримки. Загалом створено 7 таблиць, що відображають сутності у самій системі та дозволяють побудувати повноцінну систему на їх основі уникаючи компромісів у функціоналі, безпеці чи надійності.

2.3. Проєктування та реалізація алгоритмів роботи системи

Після проєктування загальної структури сайту та бази даних, необхідне проєктування алгоритмів роботи. Детально опишемо діаграму активності, зображену на рисунку 2.3:

Користувач при переході на вебзастосунок потрапляє на головну сторінку. На ній розташована навігаційна панель, завдяки якій і відбувається контроль доступу користувача до сторінок та зручна навігація.

Якщо сторінкою користується анонімний користувач, йому доступні:

- головна сторінка;
- сторінка з дописами;
- сторінка з вправами;
- сторінка із запитаннями та відповідями;
- сторінка «Підтримати»;
- можливість входу в обліковий запис.

Якщо увійшов оператор, йому стають доступні:

- база знань;
- чати;
- профіль.

Якщо користувач є авторизований з певними правами керування, йому стають доступні: сторінка профілю, сторінки адміністративної панелі відповідно до повноважень користувача. Також, за наявності доступу та відповідної кваліфікації, авторизований користувач може мати права на доступ до чатів.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

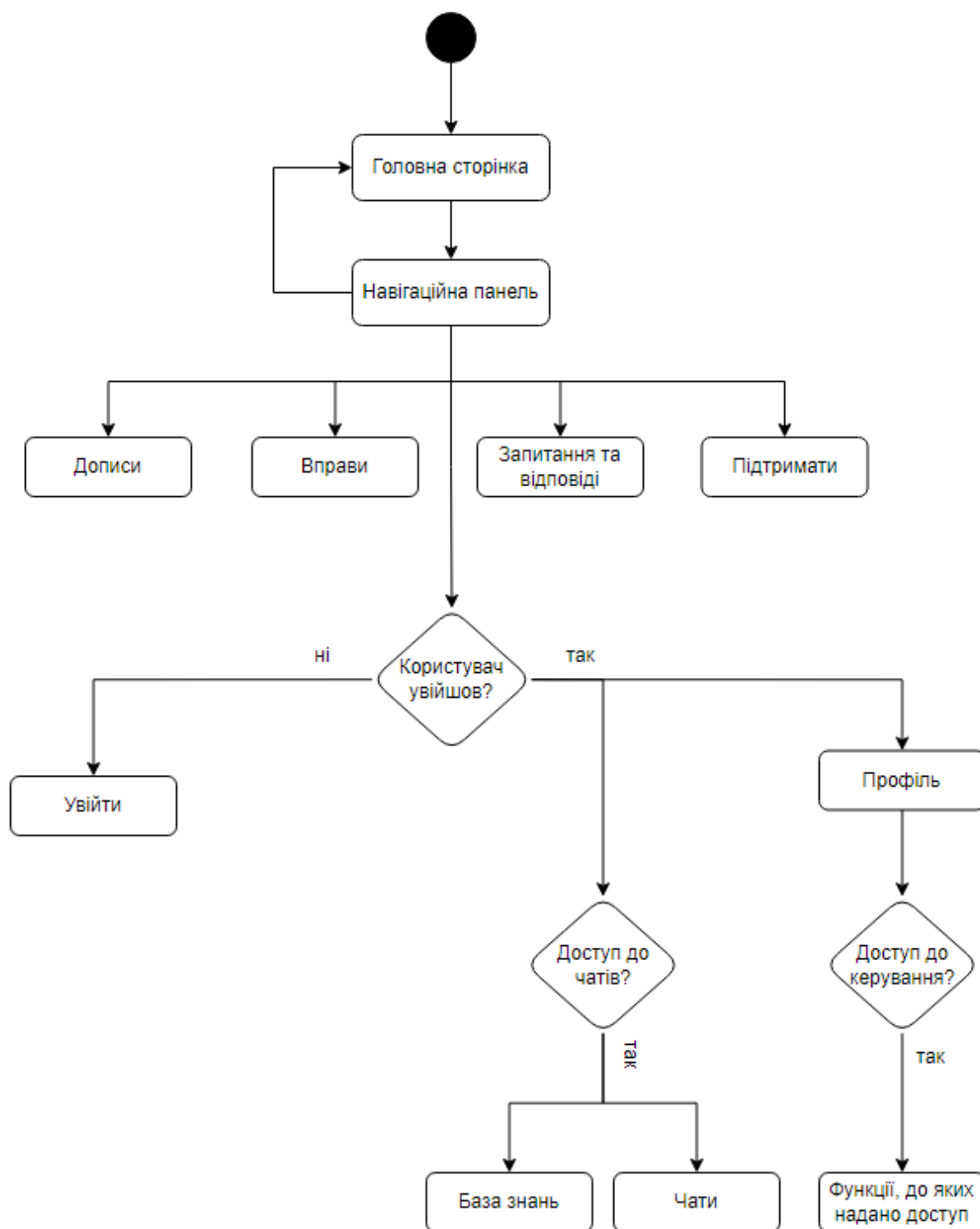


Рисунок 2.3 – Діаграма активності

2.4 Реалізація взаємодії системи та користувача

Гарним прикладом реалізації роутерів для взаємодії системи та користувача буде роутер вправ. Оскільки весь код досить об’ємний, буде наведено лише основні процедури роутера:

```

get: publicMultilevelPermissionProcedure
  .input(CUIDSchema)
  .query(async ({ input, ctx }) => {
    return ctx.db.exercise.findUnique({
      where: {
        id: input,
        ...getFullAccessConstraintWithAuthor(ctx),
      },
      include: {
        author: {
          select: { name: true },
        },
        steps: true,
      },
    });
  }),

update: multilevelPermissionProcedure
  .input(ExerciseUpdateSchema)
  .mutation(async ({ ctx, input }) => {
    const { tags, steps, ...tdi } = input;
    return ctx.db.exercise.update({
      where: {
        id: input.id,
        ...(!ctx.isFullAccess && { authorId:
ctx.session.user.id }),
      },
      data: {
        ...tdi,

        ...(tags && {
          tags: {
            connectOrCreate: tags.map((tag) => ({
              where: { title: tag },
              create: { title: tag },
            })),
          },
        }),
        ...(steps && {
          steps: {
            deleteMany: {},
            create: steps.map((step) => ({
              ...step,
              timeSeconds: step.timeSeconds
                ? Number(step.timeSeconds)
                : undefined,
            })),
          },
        }),
      },
    });
  });

```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    })),

    create: permissionProcedure
      .input(ExerciseSchema)
      .mutation(async ({ ctx, input }) => {
        const { tags, steps, ...tdi } = input;
        return ctx.db.exercise.create({
          data: {
            ...tdi,
            ...(tags && {
              tags: {
                connectOrCreate: tags.map((tag) => ({
                  where: { title: tag },
                  create: { title: tag },
                })),
              },
            )),
          },
          steps: {
            create: steps.map((step) => ({
              ...step,
              timeSeconds: step.timeSeconds
                ? Number(step.timeSeconds)
                : undefined,
            })),
          },
          author: {
            connect: { id: ctx.session.user.id },
          },
        },
      )),

    delete: multilevelPermissionProcedure
      .input(CUIDSchema)
      .mutation(async ({ ctx, input }) => {
        return ctx.db.exercise.delete({
          where: {
            id: input,
            ...(!ctx.isFullAccess && { authorId:
ctx.session.user.id }),
          },
        });
      }),
  },
);

```

Ці процедури виконують, перегляд, редагування, створення та видалення вправ. У роутері містяться і інші процедури, які дозволяють більш гнучко отримувати вправи та публікувати їх, а також створювати, оновлювати та видаляти кроки вправи.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

Значна частина створення взаємодії системи та користувача полягала у розробці інструментів для покращення ефективності та полегшення роботи операторів та інтеграції їх у чат. Такими інструментами є:

- Нотатки, які виконують функцію записника, де описано проблеми анонімного користувача та його поведінку. Даний інструмент є особливо важливим, враховуючи анонімність та відсутність доступу до історії повідомлень чату, що перейшов до іншого оператора, що є критичним для конфіденційності.
- Дописи, що дублюють відображення дописів з основної сторінки. Вони використані у чаті задля надання анонімному користувачу посилання на статтю з важливою для нього, на думку оператора, інформацією.
- Вправи, реалізація яких схожа з реалізацією дописів. Оператор, за потреби, може надіслати вправу для тимчасового покращення психологічного стану анонімного користувача.

Також варто зазначити розробку вікна помічника для оператора із використанням штучного інтелекту ChatGPT OpenAI. Його застосування дозволяє значно спростити оператору роботу з простими випадками, оскільки, за потреби, оператор може надати запит штучному інтелекту, котрий проведе швидкий аналіз чату та, на основі нього, доповість про визначений стан анонімного користувача, згенерує пропоновану відповідь та опише можливі додаткові дії. Даний інструмент також може бути використаний в якості радника у важких ситуаціях, оскільки, результат запиту до ШІ, оператор може як використати одразу, якщо він відповідає стандартам психологічних практик, так і на основі його відповіді швидко обрати напрямок думки та сформулювати свою власну.

Важливим є застереження операторів перед роботою з штучним інтелектом щодо можливих неочікуваних, невідповідних результатів, що не відповідають нормам практик психології, психіатрії та психотерапії та заклик до обережного використання згенерованих відповідей. Причиною може бути те, що

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

існуючі застосунки у сфері охорони здоров'я з використанням ІІІ часто обробляють дані з недоліками та недостатньою оцінкою ризиків. [35].

Запит до ІІІ описаний в константі AISystemMessage файлу constants.ts та приведений нижче:

```
export const AISystemMessage = {
  role: 'system',
  content:
    'Ви помічник психолога в чаті з його пацієнтами. Ви повинні надати наступне:\n' +
    'Визначити почуття пацієнта, якщо це можливо, дати фахівцю короткий аналіз повідомлень користувача (наприклад, чи є у користувача суїцидальні думки, чи він сумний, щасливий або щось інше).\n' +
    'Запропонувати тон і короткі інструкції щодо формування відповіді\n' +
    'Запропонувати лаконічну відповідь.\n' +
    'Додаткові дії, якщо вони потрібні (викликати швидку, поліцію і т.д.)\n' +
    'Якщо користувач задав подібне питання до Q&A - достатньо відповіді.\n' +
    'Ви отримаєте попередні повідомлення користувача та оператора для контексту. Відповідайте лише як помічник, не приймайте інші ролі. Розмовляйте українською.',
} as const;
```

Прикладом для демонстрації компонентів було обрано компонент Post, що відповідає за відображення сторінки з окремим дописом. У даному компоненті для відображення даних на сторінці було використано звичайні теги html, але їх стилізація була виконана із застосуванням переваг Tailwind – завдяки використанню функції сп компонент можна використати одразу в декількох місцях, наприклад, у чаті, як було згадано вище, без потреби задавати нові стилі, а лише передавши назву бажаного варіанту виконання.

Код компонента:

```
export const Post: React.FC<{
  post: RouterOutputs['post']['list']['values'][number];
  onClick?: React.MouseEventHandler<HTMLElement>;
  variant?: 'posts' | 'landing' | 'chat';
}> = ({ post, onClick, variant }) => {
  const authorName = post.author ? post.author.name : 'Unknown author';
  return (
    <article
      className={cn(
```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'flex h-full flex-col items-center gap-4 text-center
text-inherit text-slate-800 md:gap-8 md:text-left 2xl:gap-0 2xl:text-
center',
        variant === 'landing' && 'p-6 md:flex-row 2xl:flex-col',
    ))
    onClick={
        onClick
        ? onClick
        : async () => {
            await Router.push(`/posts/${post.id}`);
        }
    }
    >
    {post.image && (
        <Image
            src={post.image}
            width={1920}
            height={1080}
            alt={truncateString(post.title)}
            className={cn(
                'rounded-t-md object-contain',
                variant === 'landing' &&
                'w-2/3 rounded md:w-1/3 md:self-start 2xl:w-
full 2xl:self-auto',
            )}
        />
    )}
    <div
        className={cn(
            'flex h-full flex-col justify-between gap-2 md:gap-8
2xl:gap-0 ',
            variant === 'posts' && 'w-full',
            variant === 'chat' && 'w-full p-6',
            variant === 'landing' &&
            'items-center md:w-3/5 md:items-start 2xl:w-full
2xl:items-center',
        )}
    >
        <div
            className={cn(
                variant === 'posts' &&
                'flex w-full flex-col gap-6 p-8 text-
justify',
                variant === 'chat' &&
                'flex w-full flex-col gap-4 text-justify',
                variant === 'landing' &&
                'divide-stone-700 md:w-4/5 md:divide-y 2xl:w-
full',
            )}
        >
            <p

```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

```

3xl',
        className={cn(
            'font-bold',
            variant === 'posts' && 'md:text-xl 2xl:text-
            variant === 'landing' && 'md:pb-4 2xl:py-4',
            variant === 'chat' && 'text-xl',
        )}
    >
        {truncateString(post.title)}
    </p>
</div>
<div
    className={cn(
        'flex justify-between',
        variant === 'landing' && 'w-full',
    )}
    >
        <small
            className={cn(
                'text-xs font-light',
                variant === 'posts' && 'self-start px-8 pb-
                variant === 'landing' && 'md:self-end 2xl:mt-
            )}
            >
                {authorName}
            </small>
            <small
                className={cn(
                    'text-xs font-light',
                    variant === 'posts' && 'self-start px-8 pb-
                    variant === 'landing' && 'md:self-end 2xl:mt-
                )}
                >
                    {defaultFormatDateTime(post.createdAt)}
                </small>
            </div>
        </div>
    </article>
);
};

```

Розглянутий компонент також є репрезентацією використання типізації, що не тільки не дозволить задати неправильний варіант у подальшому використанні даного компоненту, а й дозволить середовищу розробки показати вибірку із заданих у типі за допомогою підказок.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

Також варто зауважити, що для даного проєкту була необхідна бібліотека liqpay, але, в оригінальному виконанні вона містить велику кількість недоліків та сумнівну структуру, окрім того використовує застарілі, неправильні практики валідації та не підходить для використання у даному проєкті через відсутність типізації для використання з фреймворком TypeScript. Тож було вирішено переписати оригінальний код під потреби проєкту.

Вихідний код після перебудови бібліотеки:

```
const host = 'https://www.liqpay.ua/api/';
const buttonTranslations = { uk: 'Сплатити', en: 'Pay' };

export async function api(path: string, unverifiedParams: unknown) {
  const { params, data, signature } =
    getVerifiedData(unverifiedParams);

  const dataToSend = new URLSearchParams({
    data,
    signature,
  });

  try {
    await axios.post(`${host}${params.version}/${path}`, dataToSend,
    {
      headers: {
        'Content-Type': 'application/x-www-form-urlencoded',
      },
      timeout: 5000,
      maxRedirects: 0,
    });
  } catch (error: unknown) {
    if (
      axios.isAxiosError(error) &&
      error.response &&
      error.response.status === 302 &&
      typeof error.response.headers.location === 'string'
    ) {
      return error.response.headers?.location || '';
    }
  }

  throw new Error('Failed to get response from Liqpay');
}

export function getForm(unverifiedParams: unknown = {}) {
  const { params, data, signature } =
    getVerifiedData(unverifiedParams);
```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    const buttonText = buttonTranslations[params.language];

    return `
      <form method="POST" action="https://www.liqpay.ua/api/3/checkout"
accept-charset="utf-8">
        <input type="hidden" name="data" value="${data}"/>
        <input type="hidden" name="signature" value="${signature}" />
        <script type="text/javascript"
src="https://static.liqpay.ua/libjs/sdk_button.js"></script>
        <sdk-button label="${buttonText}" background="#77CC5D"
onClick="submit()"></sdk-button>
      </form>
    `;
  }

export function getVerifiedData(unverifiedParams: unknown) {
  const params = CNBSchema.parse(unverifiedParams);
  const data = Buffer.from(JSON.stringify(params)).toString('base64');
  const signature = signString(data);
  return { params, data, signature };
}

export function signString(str: string) {
  const sha1 = crypto.createHash('sha1');
  sha1.update(env.LIQPAY_PRIVATE_KEY + str + env.LIQPAY_PRIVATE_KEY);
  return sha1.digest('base64');
}

export function getCNBObject(params: unknown) {
  const { data, signature } = getVerifiedData(params);
  return { data, signature };
}

```

Таким чином, завдяки перебудові оригінальної бібліотеки, було реалізовано можливість матеріальної підтримки проєкту у застосунку.

Висновки до другого розділу

У другому розділі було проведено аналіз основних акторів та розроблено структуру вебзастосунку для психологічної підтримки. Ця частина роботи включала огляд взаємодії користувачів із системою та обґрунтування функціональних вимог до застосунку. На основі даного аналізу була розроблена комплексна структура вебзастосунку, яка забезпечує зручний та ефективний доступ до різноманітних ресурсів.

Особливу увагу було приділено розробці бази даних системи психологічної підтримки. Була спроектована гнучка та масштабована база даних,

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

яка здатна забезпечити зберігання та швидкий доступ до великої кількості даних, необхідних для функціонування системи. Це включає вправи, дописи, дані про користувачів, повідомлення, чати, та інші критично важливі дані.

Проектування та реалізація основних алгоритмів роботи системи дозволили передбачити поведінку користувача, переглянути, доступ до яких об'єктів необхідний групам користувачів, та доступ до чого потрібно обмежити та реалізувати це в системі.

Окремо була розглянута реалізація взаємодії системи та користувача, включаючи розробку роутерів та компонентів, що забезпечують інтуїтивно зрозумілий та зручний інтерфейс. Було здійснено перебудову бібліотеки оплати, що забезпечує безпечні та надійні фінансові транзакції, підтримуючи різноманітні способи оплати для користувачів.

Узагальнюючи, другий розділ детально охопив всі аспекти розробки взаємодії системи користувача на прикладі вебзастосунку для психологічної підтримки, від аналізу та структурування до технічної реалізації.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

3. ІНТЕРФЕЙС ТА ПОРЯДОК ВЗАЄМОДІЇ З СИСТЕМОЮ ЗАСТОСУНКУ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ

3.1 Порядок встановлення та налаштування системи

Розроблена система має два основних режими роботи: режим розробки та режим повної експлуатації, кожен з яких володіє певними характеристиками та призначений для виконання чітко окреслених завдань.

Режим розробки орієнтований на полегшення процесу розробки та тестування програмного забезпечення і характеризується монолітною архітектурою, що спрощує розуміння та модифікацію основного програмного коду. Завдяки Next.js сервер миттєво перезапускається після внесення змін до коду, дозволяючи розробнику негайно спостерігати за результатами. Застосунок працює на нестандартному 80 порті, полегшуючи тестування за наявності доступу до комп'ютера з глобальної мережі. Сторінки системи доповнені спеціальними компонентами для налагодження, які візуалізують помилки, перезавантажують сторінку після змін до коду та автоматично відновлюють з'єднання при перезапуску сервера.

Режим повної експлуатації орієнтований на забезпечення повноцінної роботи системи в реальних умовах і характеризується мікросервісною архітектурою, яка робить систему більш масштабованою та стійкою до збоїв. Процес розгортання системи для роботи в режимі повної експлуатації детально описаний у вигляді діаграми розгортання, яка візуалізує взаємозв'язки між компонентами системи та демонструє їх розташування на серверній інфраструктурі.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

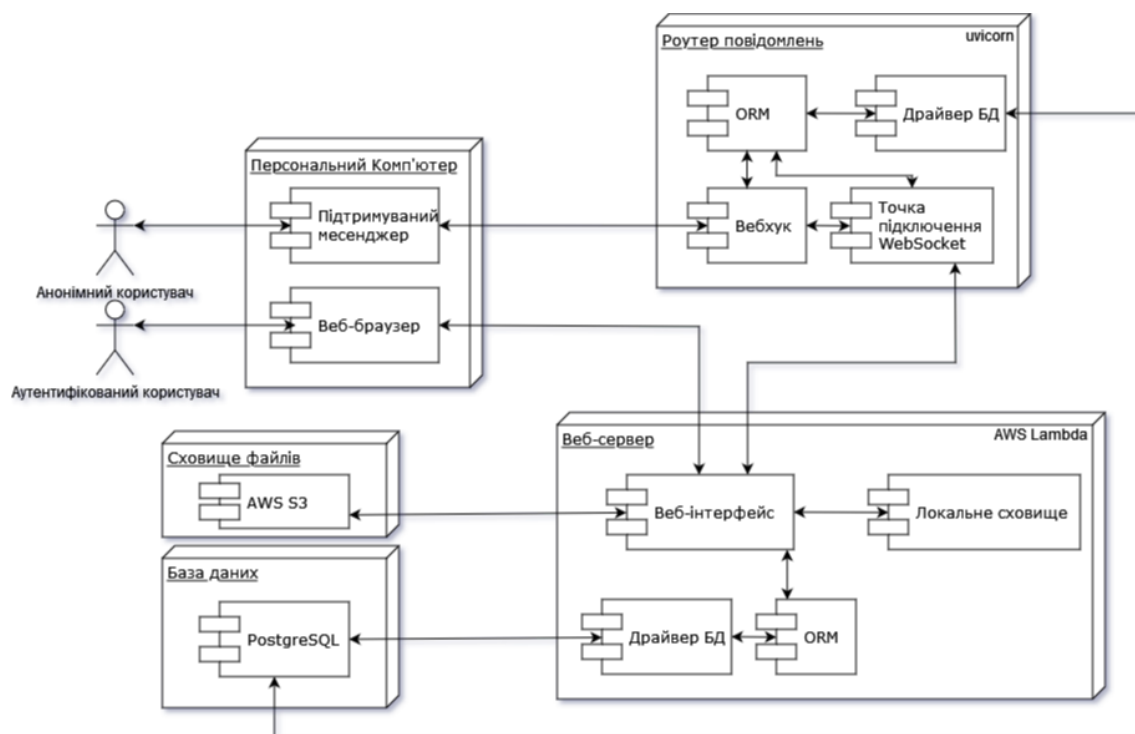


Рисунок 3.1. Діаграма розгортання застосунку.

На рисунку 3.1 представлена структура компонентів системи після розгортання в реальному середовищі, що суттєво відрізняється від локального запуску. Замість монолітного коду та основний застосунок розгорнуті як мікросервіси на платформі AWS Lambda, що робить систему більш масштабованою, стійкою до збоїв та простішою у керуванні. У режимі повної експлуатації застосунки позбавлені функцій, орієнтованих на спрощення розробки, таких як фіксація змін, автоматичне перезавантаження та детальне відображення помилок, що значно підвищує їх продуктивність та зменшує час відгуку.

Діаграма окрім основних компонентів також відображає базу даних, до якої підключаються як основний застосунок, так і віддалене сховище файлів, де зберігаються медіафайли користувачів.

Вебзастосунок передбачає два типи користувачів: анонімний, тобто користувач, якому доступна загальна інформація, можливість підтримки проєкту та посилання на чати, та аутентифікований користувач, який отримує доступ до функцій в залежності від наданих йому дозволів. Для автоматизації процесу розгортання системи використовується GitLab CI/CD pipeline, що автоматизує

процес збірки коду, тестування та публікації його в реальному середовищі, з конфігурацією GitLab Runner за допомогою відповідного файлу:

```
stages:
  - deploy

deploy_preview:
  stage: deploy
  except:
    - main
  script:
    - vercel deploy --yes --token=$VERCEL_TOKEN

deploy_production:
  stage: deploy
  only:
    - main
  script:
    - vercel deploy --yes --prod --token=$VERCEL_TOKEN
```

Завдяки використанню сервісу Vercel, конфігурацію розгортання зменшено до мінімуму – одного запиту до сервісу з секретним ключем.

3.2. Опис інтерфейсу програмного продукту

В першу чергу для зацікавлення та утримання цільової аудиторії, було обрано кольорову гаму нейтральних кольорів, обрано легкий, ненав'язливий дизайн та шрифт і колір для логотипу. Усе це можна спостерігати на головній сторінці застосунку.



Рисунок 3.2. Загальна інформація.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк. 55
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		

На рисунку 3.2 відображено логотип, інформацію щодо проекту та посилання на чати у різних месенджерах для швидкого доступу до отримання психологічної підтримки кваліфікованим спеціалістом.

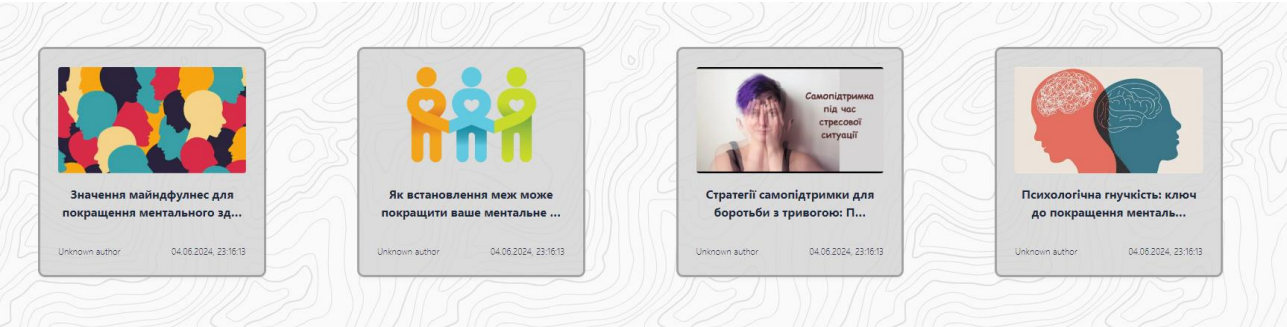


Рисунок 3.3. Відображення останніх дописів.

На рисунку 3.3 видно вигляд останніх доданих дописів, що відображаються на головній сторінці. Разом з ними також на головній сторінці розміщені блоки «Чим ми надихались» та «Як ми можемо допомогти».

На головній сторінці також розміщені компоненти Header та Footer, що були об’єднані в компонент Layout, разом з основною стилізацією вмісту.

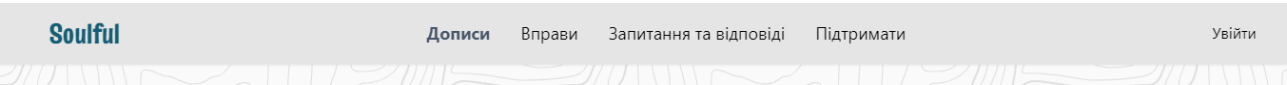


Рисунок 3.4. Шапка вебзастосунку.

Компонент Header (рисунок 3.4) містить логотип, що виконує функцію оформленого посилання на головну сторінку, навігаційну панель, елементи якої виділяються, якщо користувач перебуває на відповідній сторінці, та кнопку «Увійти» або компонент, що відображає профіль користувача.



Рисунок 3.5. Підвал сайту.

На компоненті Footer (рисунок 3.5) розміщені контакти для зв’язку, продубльовано посилання для навігації по сайту та посилання на чати з головної сторінки.

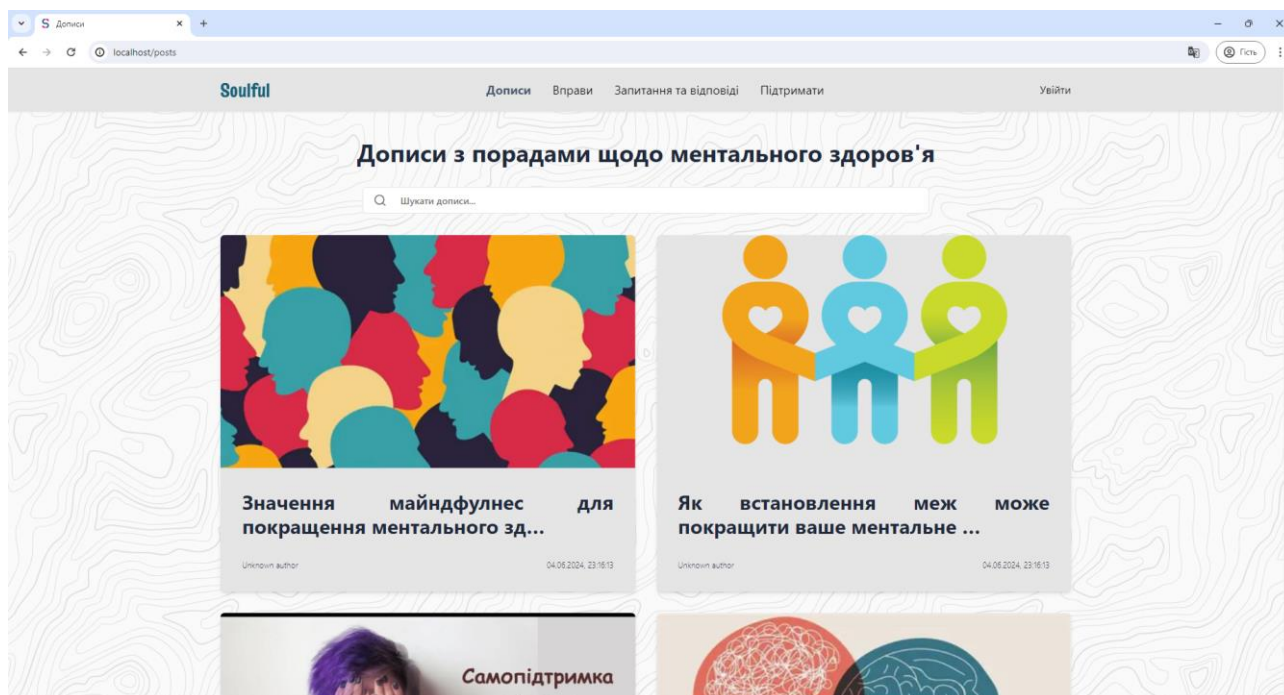


Рисунок 3.6. Сторінка з дописами.

На рисунку 3.6. зображено сторінку, на якій містяться інформаційні матеріали та статті щодо різних аспектів психологічного здоров'я.

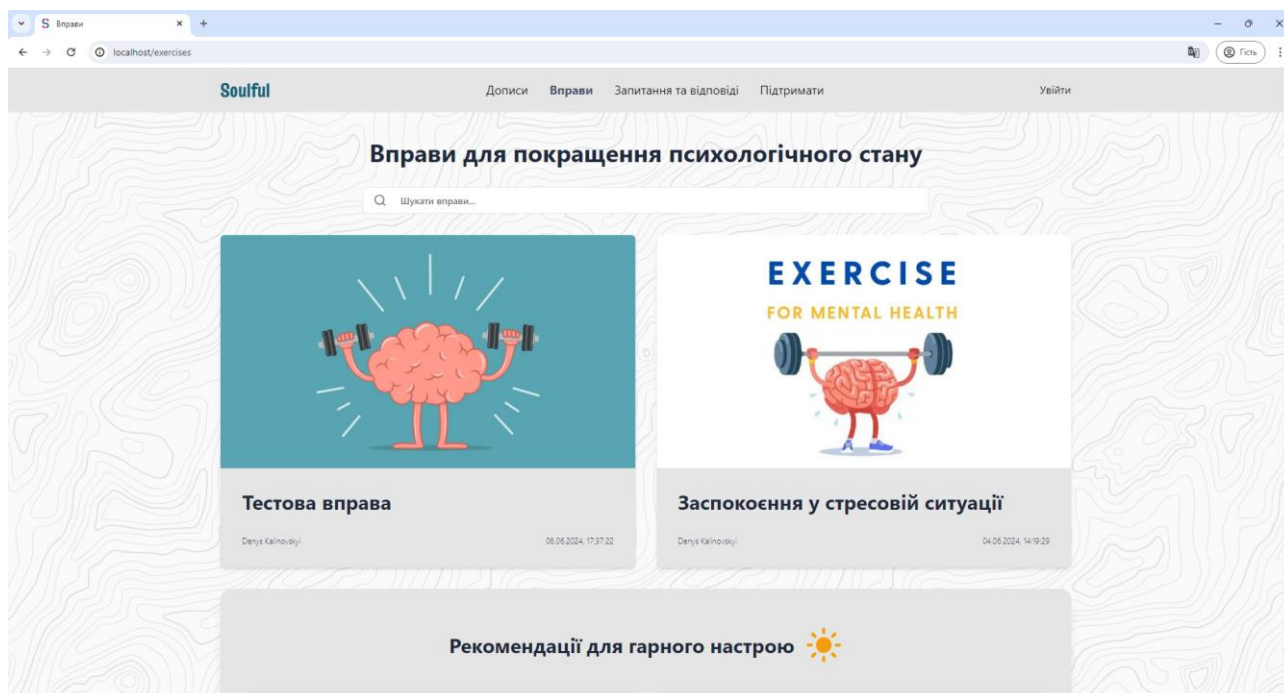


Рисунок 3.7. Сторінка з вправами.

Також у застосунку реалізовані вправи для покращення психологічного стану, сторінку з якими можна розглянути на рисунку 3.7.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				57
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.8. Реалізація пошуку.

Для зручності користування було реалізовано пошук по вправах та дописах, демонстрацію якого видно на рисунку 3.8.

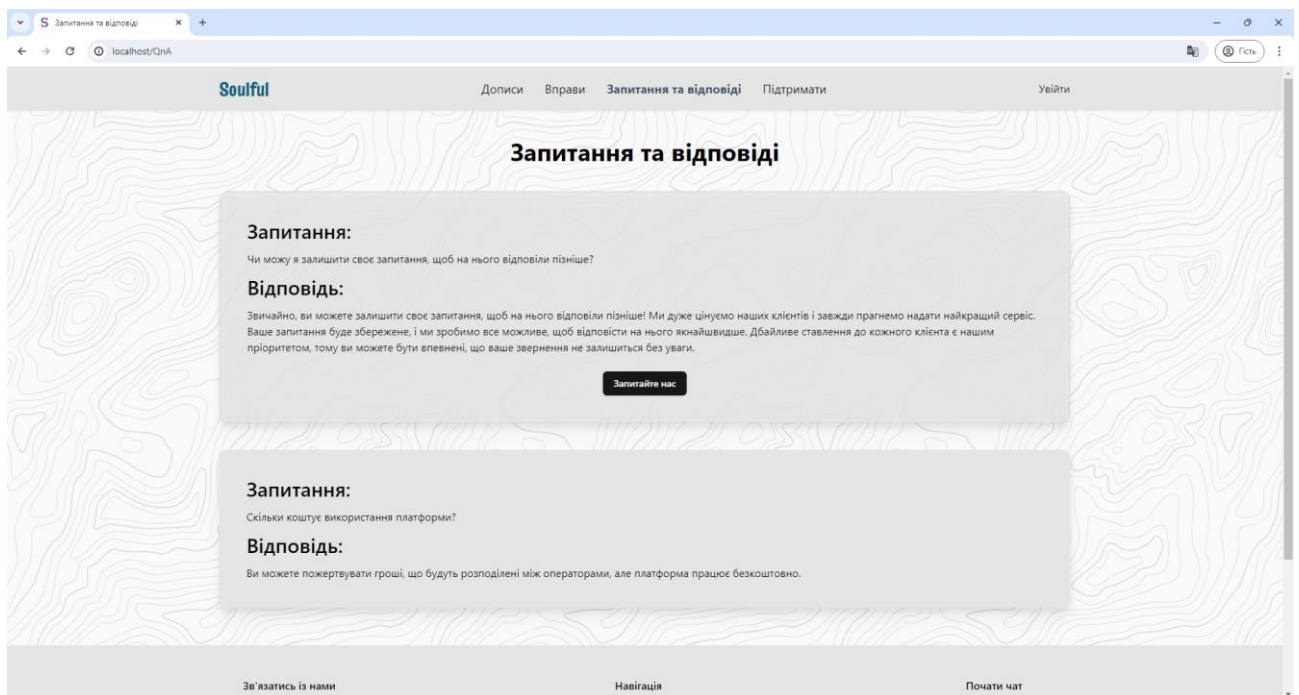


Рисунок 3.9. Сторінка «Запитання та відповіді».

Було реалізовано також і можливість залишити свої запитання щодо роботи проєкту, освіти спеціалістів і так далі. З цією метою створено сторінку, зображену на рисунку 3.9. Після натискання на кнопку «Запитайте нас», користувач бачить форму, вказану на рисунку 3.10.

		Чернюк А.М.,			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

я, щоб на нього відповіли пізніше?

Задати питання

Ім'я

Антон

Ел. пошта

anton@gmail.com

Запитання

Чи можливо...

Підтвердити

є запитання
ми зробим
впевнені, і
завжди прагне
влення до кож
норми?

Рисунок 3.10. Форма для запитання.

Після залишення заявки з запитанням, відповідь користувачу може бути надана як у вигляді листа на залишену електронну пошту, так і відредагувавши запитання, авторизований користувач може опублікувати його на сторінці за допомогою адміністративної частини.

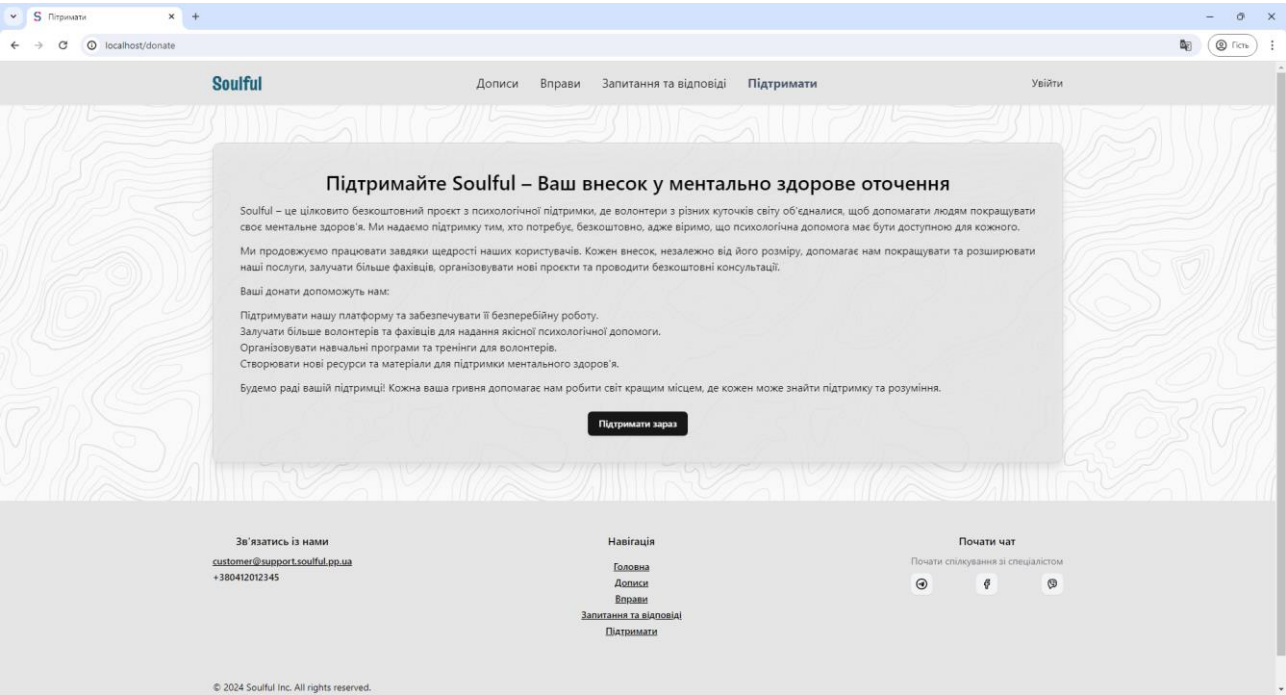


Рисунок 3.11. Сторінка «Підтримати».

Оскільки проєкт створений під використання його як волонтерського, реалізовано сторінку з можливістю підтримати проєкт фінансово. Після натискання на кнопку «Підтримати зараз», буде відкрито форму, зображену на рисунку 3.12.

Рисунок 3.12. Форма для підтримки.

Після вибору типу підтримки та внесення суми, користувач зможе виконати переказ коштів за допомогою сторінки liqpay. Після успішної оплати для висловлення щирої подяки та заохочення підтримки надалі, було створено сторінку сторінку (рисунок 3.13).

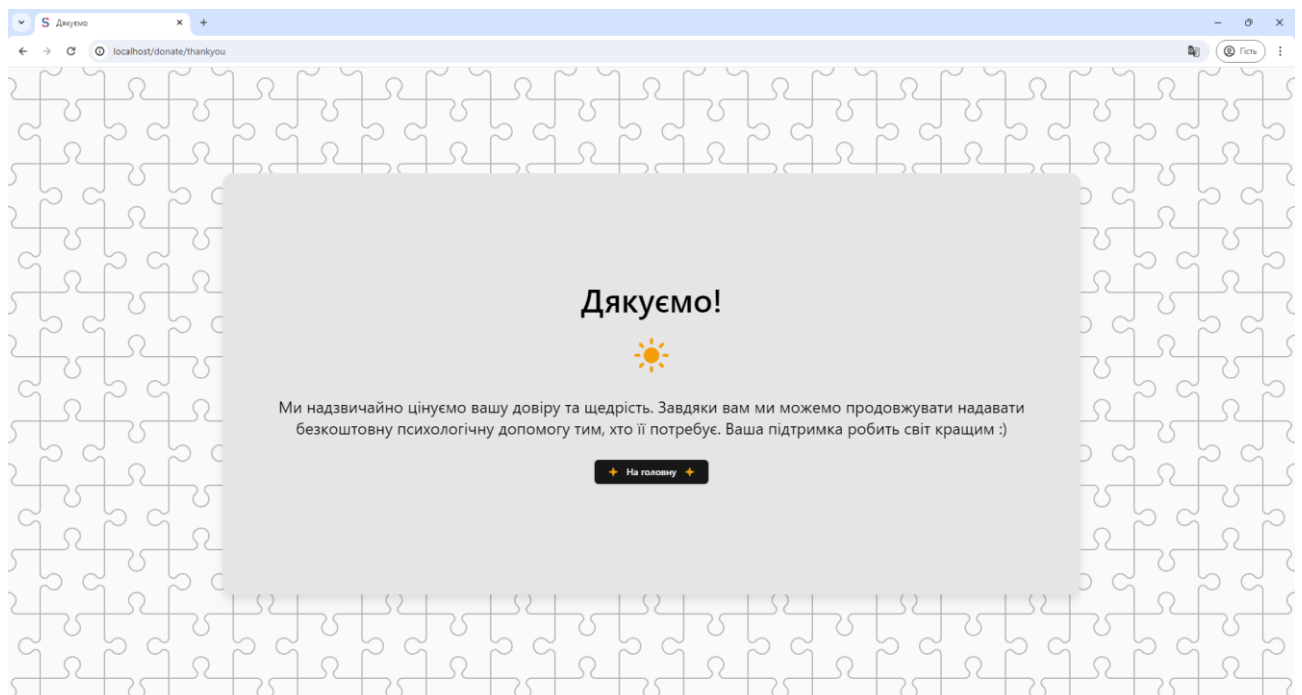


Рисунок 3.13. Подяка за підтримку проєкту.

З метою висловлення щирості до користувача, що важливо для, можливо, вразливого користувача, було розроблено компонент з короткими рекомендаціями щодо ментального здоров'я (рисунок 3.14), що відображаються у довільному порядку поміж вправ та дописів.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

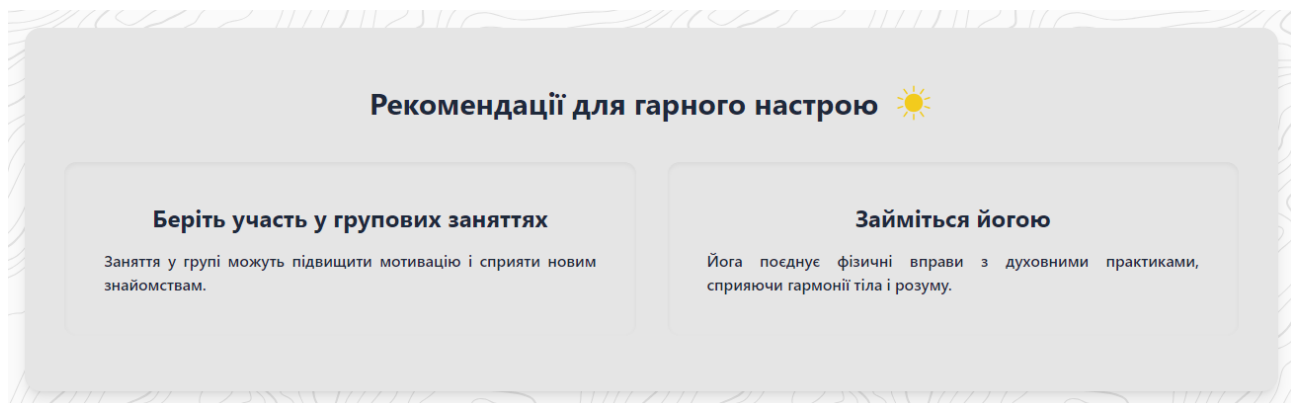


Рисунок 3.14. Рекомендації для гарного настрою.

Також було розроблено сторінку входу в обліковий запис (рисунок 3.15). Сторінка реєстрації для даного застосунку не є необхідною, оскільки реєстрація операторів відбувається за допомогою адміністративної панелі.

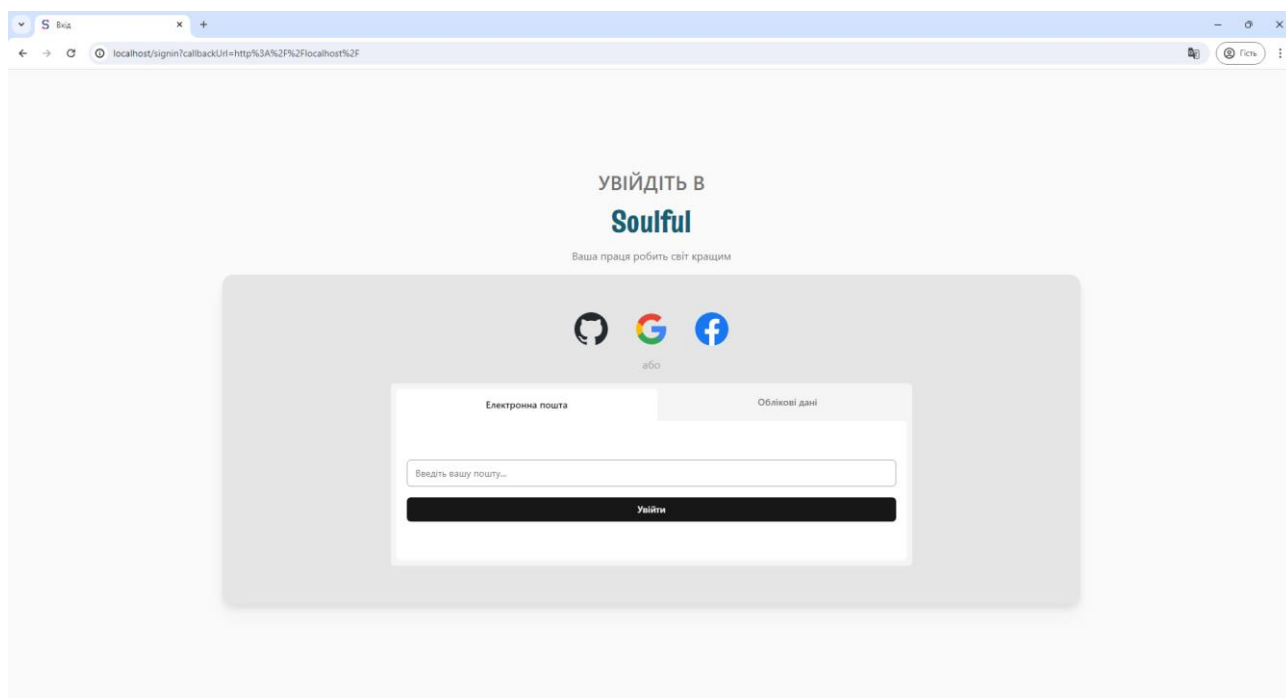


Рисунок 3.15. Сторінка входу.

Акцент даної сторінки входу знаходиться на вході за допомогою інших сервісів через їх зручність та можливість їх реалізації в NextAuth.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

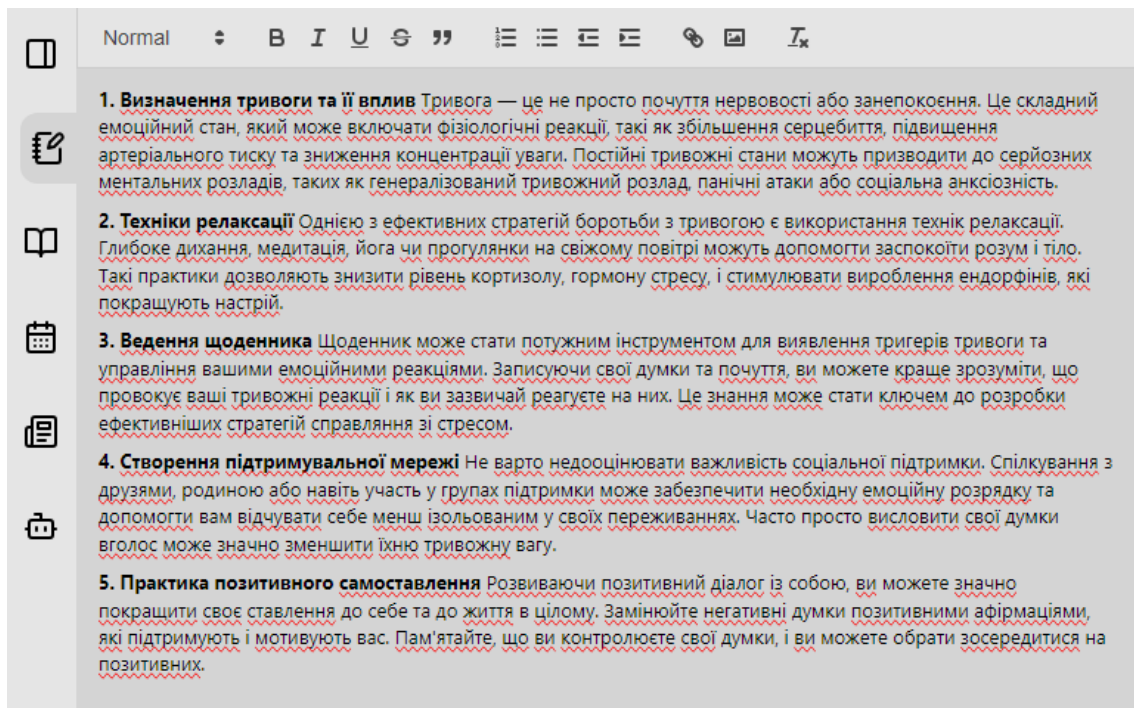


Рисунок 3.16. Нотатки до користувача.

Однією з найважливіших для зручності у використанні чатів операторами та конфіденційності розмов між операторами та анонімними користувачами, є реалізація нотаток (рисунок 3.16) на бічній панелі, прикріплених до певного чату. Для розуміння загального вигляду в контексті всієї сторінки, додано рисунок 3.17.

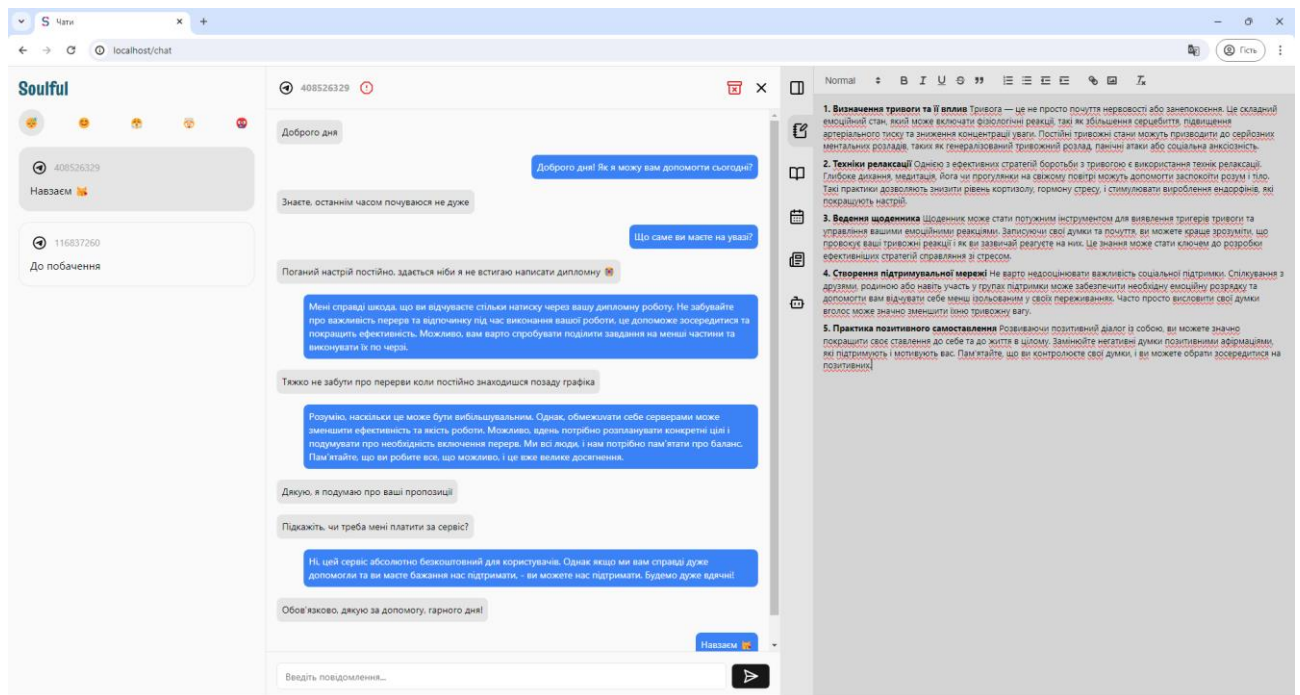


Рисунок 3.17. Загальний вигляд вікна чату з нотатками.

		Чернюк А.М.,				Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

Якщо історія чатів при повторному зверненні не зберігається, то нотатки, що закріплені за певною особою, дозволяють надавати інформацію про попередні звернення операторам, до яких користувач потрапить при повторному зверненні.

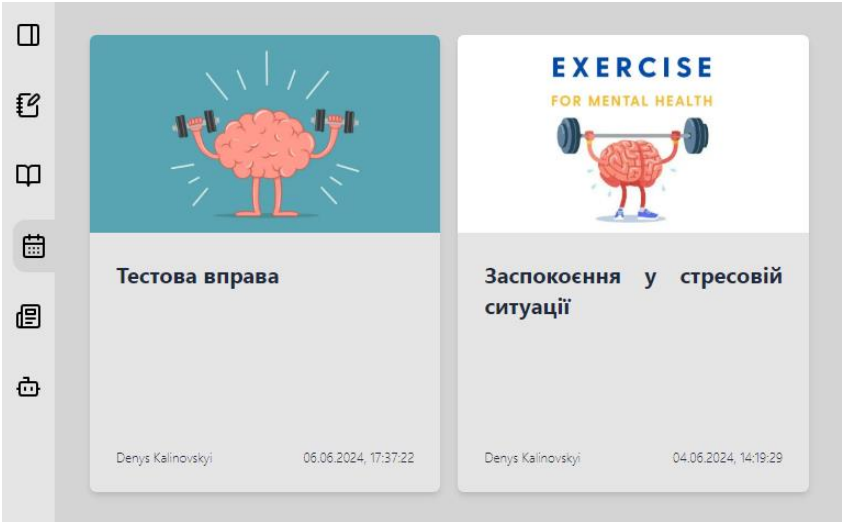


Рисунок 3.18. Вправи у чаті.

Для операторів було додано можливість переглядати список вправ безпосередньо у чаті (рисунок 3.18) та, при натисканні на них, додавати посилання на вправи у чат.

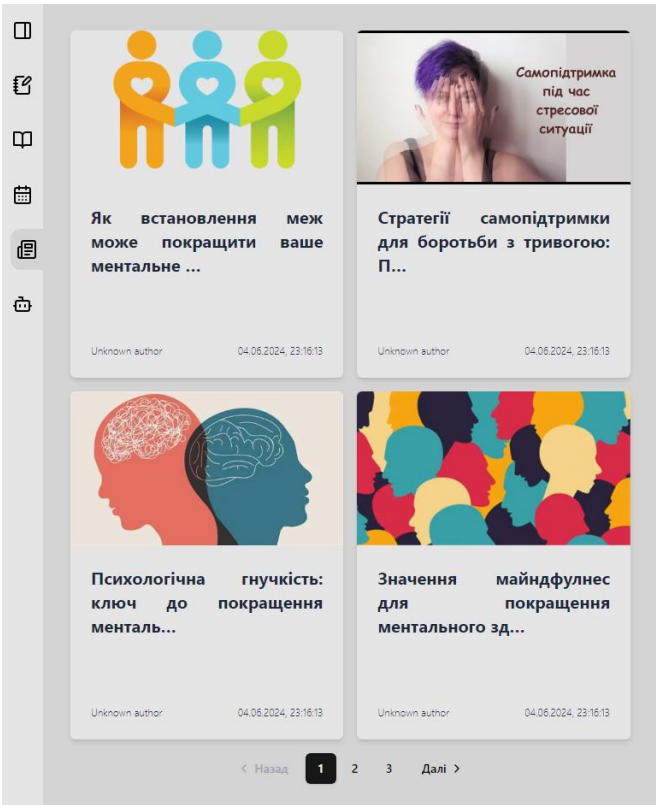


Рисунок 3.19. Дописи у чаті

Таким же чином було реалізовано можливість перегляду дописів (рисунок 3.19)

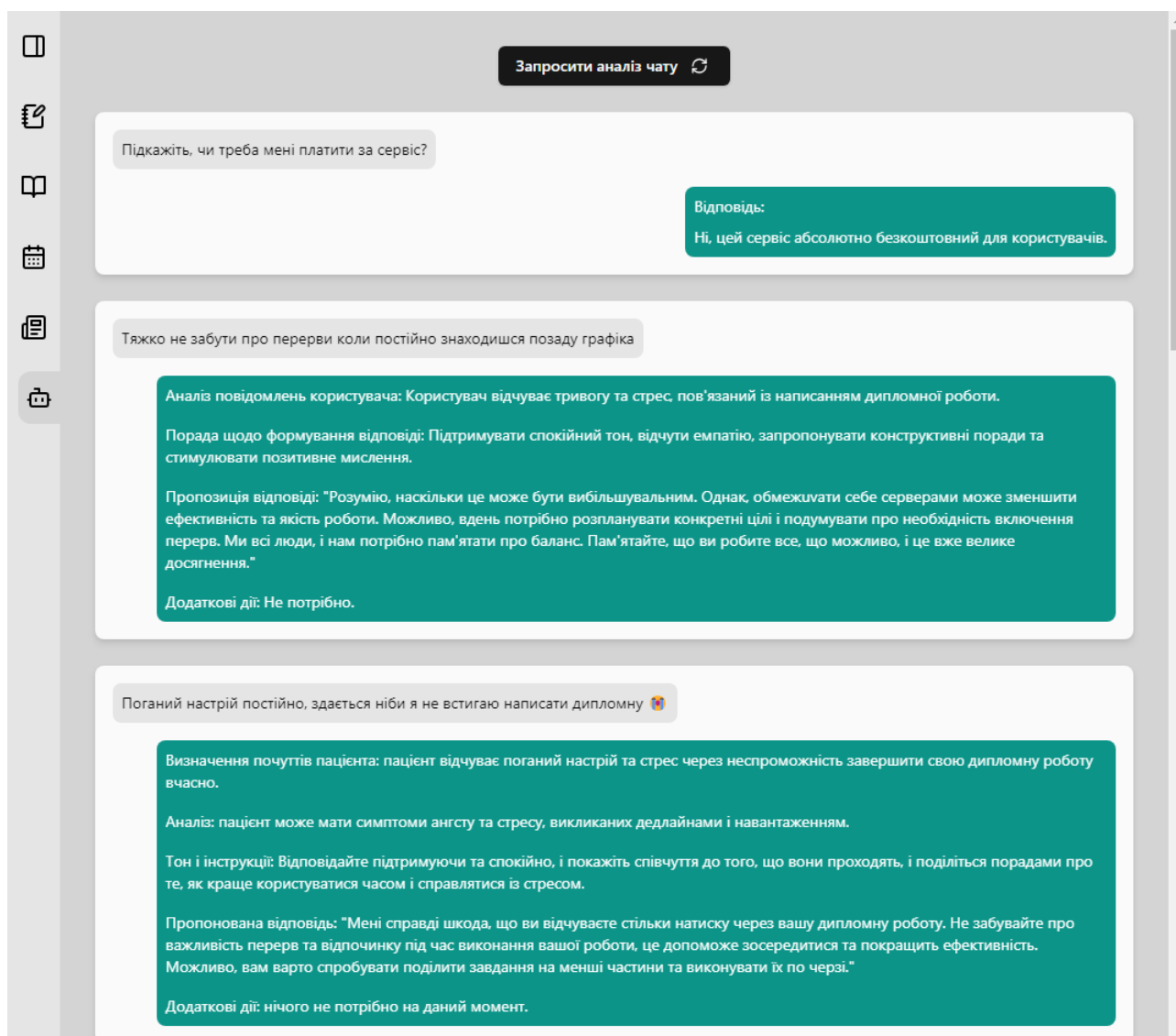


Рисунок 3.20. Вікно аналізу чату штучним інтелектом.

Штучний інтелект у чаті було реалізовано з використанням приємного дизайну (рисунок 3.20), з оформленням останнього повідомлення анонімного користувача та відповіді ШІ у вигляді повідомлень. Згенерувати новий аналіз користувач може за натисканням кнопки «Запросити аналіз чату».

Таким чином, інтерфейс взаємодії системи користувача виконаний з урахуванням потреб щодо дизайну та досвіду використання.

3.3. Тестування застосунку психологічної підтримки.

У роботі застосунку критичною є стабільність та відсутність помилок у роботі. Для контролю високої якості створеного програмного забезпечення проведено тестування, випадки якого наведено в таблиці 3.1.

Таблиця 3.1

Тестові випадки

Опис	Кроки	Вхідні дані	Очікуваний результат	Фактичний результат	Результат
1	2	3	4	5	6
Перехід за посиланням з головної сторінки	Відкрити сайт та натиснути на логотип месенджера	-	Перехід на сторінку чату психологічної підтримки у відповідному месенджері	Перехід на сторінку чату психологічної підтримки у відповідному месенджері	Успіх
Перегляд дописів та вправ	Перейти на головну сторінку вебзастосунку. Натиснути на посилання до відповідної сторінки на навігаційній панелі, Клікнути по необхідному допису або вправі	-	Відображення відповідної статті чи вправи на сторінці.	Відображення відповідної статті чи вправи на сторінці.	Успіх
Пошук дописів та вправ	Перейти на головну сторінку вебзастосунку. Натиснути на посилання до відповідної сторінки на навігаційній панелі. Ввести в поле пошуку ключове слово.	Слово, що міститься у назві або вмісту допису або вправи.	Відображення на сторінці дописів або вправ тільки сутностей, що містять ключове слово.	Відображення на сторінці дописів або вправ тільки сутностей, що містять ключове слово.	Успіх

Продовження таблиці 3.1.

1	2	3	4	5	6
Видалення вправи або допису	1. Перейти на головну сторінку вебзастосунку. 2. Натиснути на посилання до відповідної сторінки на навігаційній панелі. 3. Клікнути по необхідному допису або вправі. 4. Натиснути на кнопку «Видалити».	Права для видалення дописів та вправ.	Відкритий допис або вправу, за наявності необхідних для цього прав, повинно бути видалено.	Користувач видалив допис, оскільки мав необхідні права.	Успіх
Розділення дописів та вправ на сторінки	1. Перейти на головну сторінку вебзастосунку. 2. Натиснути на посилання до відповідної сторінки на навігаційній панелі. 3. Прогорнути сторінку до низу.	Наявність більше 8 дописів або вправ	Відображення елемента, що виконує функцію розділення вмісту на сторінки та навігації між ними.	Відображення елемента, що виконує функцію розділення вмісту на сторінки та навігації між ними.	Успіх
Можливість підтримки проєкту	1. Перейти на головну сторінку вебзастосунку. 2. Натиснути на посилання «Підтримати» в навігаційній панелі. 3. Натиснути кнопку «Підтримати зараз». 4. Обрати тип допомоги. 5. Ввести суму. 6. Натиснути кнопку «Оплатити».	Сума фінансової підтримки.	Перенаправлення на сторінку оплати та, після успішного її проведення, відкриття сторінки подяки за підтримку проєкту.	Перенаправлення на сторінку оплати та, після успішного її проведення, відкриття сторінки подяки за підтримку проєкту.	Успіх

Продовження таблиці 3.1.

1	2	3	4	5	6
Запит до ШІ в чаті	1. Перейти на головну сторінку вебзастосунку. 2. Клікнути на посилання «Чат» в навігаційній панелі. 3. Відкрити активний чат. 4. Відкрити бічну панель. 5. Відкрити вкладку з логотипом бота. 6. Натиснути «Запросити аналіз чату».	Права на доступ до чату та активний чат, прикріплений до оператора з повідомленням і від анонімного користувача.	Генерація аналізу чату та відповідь на запит, оформлена у блок повідомлень, у якому штучний інтелект відповідає на останнє повідомлення від анонімного користувача	Генерація аналізу чату та відповідь на запит, оформлена у блок повідомлень, у якому штучний інтелект відповідає на останнє повідомлення від анонімного користувача	Успіх
Збереження нотаток до анонімного користувача в чаті	1. Перейти на головну сторінку вебзастосунку. 2. Клікнути на посилання «Чат» в навігаційній панелі. 3. Відкрити активний чат. 4. Відкрити бічну панель. 5. Відкрити вкладку нотаток. 6. Ввести нотатки щодо користувача. 7. Перезавантажити сторінку.	Права на доступ до чату та активний чат, прикріплений до оператора.	Відображення сповіщення «нотатки збережені» після затримки протягом хвилини після написання тексту. Відображення змінених нотаток після перезавантаження сторінки.	Відображення сповіщення «нотатки збережені» після затримки протягом хвилини після написання тексту. Відображення змінених нотаток після перезавантаження сторінки.	Успіх

Продовження таблиці 3.1.

1	2	3	4	5	6
Авторизація користувачі в	1. Перейти на головну сторінку вебзастосунку. 2. Клікнути на посилання «Увійти» в правій частині шапки сайту. 3. Натиснути на бажаний варіант входу для авторизації (GitHub, Google або Facebook).	Наявний обліковий запис у GitHub, Google або Facebook	Користувача авторизовано за допомогою обраного облікового запису.	Користувача авторизовано за допомогою обраного облікового запису.	Успіх

Додатково проведено повне тестування системи користувачами, в ході якого виявлено 5 невеликих недоліків.

Було проведено мануальне тестування критично важливого функціоналу. Виявлені недоліки були виправлені та додані у фінальне оновлення застосунку. Також виконано повторне тестування, яке не виявило жодних додаткових помилок. Відповідно, тестування завершено успішно.

У рамках третього розділу проведено детальний аналіз ключових аспектів, що впливають на стабільність та ефективність системи адміністрування застосунку психологічної підтримки, розглянуто два основних режими роботи: локальний режим розробки та повноцінний режим експлуатації. Також розглянуто реалізацію автоматичного розгортання за допомогою GitLab CI/CD pipeline.

Також було детально описано створення головної сторінки, сторінок дописів, вправ, авторизації та підтримки проєкту. Значну увагу приділено інтеграції штучного інтелекту, який використовується як помічник для оператора в чаті психологічної підтримки та для створення нотаток, прикріплених до анонімного користувача. Результати проведеного тестування свідчать про успішну реалізацію ключових функцій, що забезпечують зручність користування, інтерактивність і надійність вебзастосунку.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В ході розробки взаємодії системи та користувача, було досліджено сучасні підходи та методики взаємодії людини з інформаційно-комунікаційною системою. Метою роботи було створення застосунку психологічної підтримки, що надає користувачам можливість використання його як електронного ресурсу для покращення психологічного стану, а також створення інструментів для полегшення взаємодії операторів з анонімними користувачами у чаті.

Значимість даного проєкту полягає в тому, що він відкриває нові можливості для покращення психологічного здоров'я та забезпечує доступ користувачів до якісної психологічної підтримки у віддаленому режимі. Реалізація даного проєкту забезпечує надання психологічної допомоги, що може виявитися критичним у сучасному світі, особливо в умовах пандемії, коли питання психічного здоров'я стає надзвичайно актуальним.

Впровадження даного застосунку сприяє покращенню якості життя користувачів шляхом забезпечення доступу до якісної психологічної підтримки та зручної взаємодії з операторами чату. Також він відкриває можливості для подальших досліджень у сфері психологічної допомоги через інформаційні технології.

Робота підтверджує можливість успішного поєднання сучасних технологій з психологічною практикою для досягнення позитивного впливу на психічне здоров'я людей.

У загальному висновку слід зазначити, що розроблена система психологічної підтримки на основі взаємодії системи та користувача виявляється рішенням, яке відкриває нові перспективи для надання психологічної допомоги в сучасному світі. Впровадження цього застосунку сприятиме покращенню психологічного здоров'я та якості життя користувачів, підвищить ефективність взаємодії з операторами чату, а також відкриє нові горизонти для подальших досліджень у галузі психологічної допомоги через інформаційні технології.

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чернюк А.М., Каліновський Д.Ю., Савіцький Р.С. ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В СИСТЕМАХ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ. Тези XV міжнародної науково-практичної конференції «Комп'ютерні системи та мережні технології», 25 – 26 квітня 2024. Київ: «НАУ», 2024. С. 169.
2. Каліновський Д.Ю., Чернюк А.М., Савіцький Р.С. ІНТЕГРАЦІЯ МУЛЬТИПЛАТФОРМНИХ МЕСЕНДЖЕРІВ З ВИКОРИСТАННЯМ FASTAPI ТА WEBSOCKETS ДЛЯ СТВОРЕННЯ УНІФІКОВАНОЇ КОМУНІКАЦІЙНОЇ ПЛАТФОРМИ. Тези XIV Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології – 2024», 28-29 березня 2024 р. Житомир: ДУ «Житомирська Політехніка», 2024. С. 13.
3. Hedery: Спробуйте онлайн психотерапію [Електронний ресурс] - Режим доступу до ресурсу: <https://hedery.com.ua/>
4. Розкажи мені. Ваша підтримка поруч [Електронний ресурс] – Режим доступу до ресурсу: <https://tellme.com.ua/>
5. Тепло – психолог онлайн [Електронний ресурс] – Режим доступу до ресурсу: <https://teplo.app/>
6. JavaScripts documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
7. JavaScript in web development [Електронний ресурс] – Режим доступу до ресурсу: <https://groovetechnology.com/blog/javascript-in-web-development-what-is-javascript-understanding-the-importance-and-uses/>
8. React documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://react.dev/reference/react>
9. Virtual DOM in React [Електронний ресурс] – Режим доступу до ресурсу: <https://drukarnia.com.ua/articles/virtual-dom-in-react-Au-O->
10. Create T3 App template documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://create.t3.gg/en/introduction>

		Чернюк А.М.,			ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
		Савіцький Р.С.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Why I choose T3-stack? [Електронний ресурс] – Режим доступу до ресурсу:
<https://dev.to/zenstack/why-i-choose-t3-stack-as-the-fullstack-to-build-the-react-app-1e1k>
12. Next.js documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://nextjs.org/docs>
13. Key features of Next.js [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.linkedin.com/pulse/key-features-nextjs-sajjad-hossain-fx0qc/>
14. Tailwindcss documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://tailwindcss.com/docs/>
15. CSS documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://devdocs.io/css/>
16. TypeScript documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.typescriptlang.org/docs/>
17. Benefits of TypeScript [Електронний ресурс] – Режим доступу до ресурсу:
<https://startup-house.com/blog/benefit-of-typescript>
18. Prisma documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.prisma.io/docs>
19. Discover the benefits of using prisma [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.brilworks.com/blog/discover-the-benefits-of-using-prisma-in-your-next-project/>
20. NextAuth.js Documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://next-auth.js.org/getting-started/introduction>
21. User authentication in Next.js app using NextAuth [Електронний ресурс] – Режим доступу до ресурсу:
<https://nextjsdev.com/user-authentication-in-nextjs-13-app-using-nextauth/>
22. tRPC documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://trpc.io/docs/>
23. What is tRPC [Електронний ресурс] – Режим доступу до ресурсу:
<https://reliasoftware.com/blog/what-is-trpc>

- 24.Zod documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://zod.dev/>
- 25.Radix UI Documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.radix-ui.com/primitives/docs/overview/introduction>
- 26.Shadcn UI Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://ui.shadcn.com/docs>
- 27.Radix UI adoption guide [Електронний ресурс] – Режим доступу до ресурсу:
<https://blog.logrocket.com/radix-ui-adoption-guide/>
- 28.Shadcn UI and Material UI comparison [Електронний ресурс] – Режим доступу до ресурсу: <https://djangostars.com/blog/shadcn-ui-and-material-design-comparison/>
- 29.AWS SDK for JavaScript v3 Documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.aws.amazon.com/AWSJavaScriptSDK/v3/latest/>
- 30.AWS CLI v2 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.aws.amazon.com/cli/>
- 31.VSCode documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://code.visualstudio.com/docs>
- 32.PostgreSQL 16.3 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/docs/16/index.html>
- 33.PgAdmin Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pgadmin.org/docs/>
- 34.OpenAI API Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://platform.openai.com/docs/overview>
- 35.AI In Mental Health [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.forbes.com/sites/bernardmarr/2023/07/06/ai-in-mental-health-opportunities-and-challenges-in-developing-intelligent-digital-therapies/>

		Чернюк А.М.,			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савицький Р.С.				72
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІЗ	Арк.
		Савіцький Р.С.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

SignIn custom page

```

export const SignIn = ({
  csrfToken,
}: InferGetServerSidePropsType<typeof getServerSideProps>) => {
  const { data: session } = useSession();

  const router = useRouter();

  useEffect(() => {
    if (session) {
      void router.push('/');
    }
  }, [session]);

  return (
    <>
      <Head>
        <title>Вхід</title>
      </Head>
      <div className="flex h-svh w-full flex-col items-center
justify-center gap-4">
        <h2 className="font-semibold uppercase text-neutral-500">
          Увійдіть в
        </h2>
        <Logo className="text-5xl" />
        <p className="text-neutral-500">
          Ваша праця робить світ кращим
        </p>
        <div className="flex w-2/3 flex-col items-center justify-
center gap-4 rounded-2xl bg-neutral-200 p-14 drop-shadow-xl">
          <div className="flex w-2/3 justify-center gap-10">
            <Button
              onClick={async () => {
                await signIn('github', { callbackUrl:
'/' });
              }}
              variant="ghost"
              className="h-[50px] w-[50px] rounded-full p-
0"
            >
              <Image
                src={'/images/github-mark.svg'}
                className="rounded-full"
                width={50}
                height={50}
                style={{
                  aspectRatio: '1',
                  objectFit: 'cover',
                }}
              />
            </Image>
          </div>
        </div>
      </div>
    </>
  );

```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савицький Р.С.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        alt="LogIn by github"
      />
    </Button>
    <Button
      onClick={async () => {
        await signIn('google', { callbackUrl:
'/' });

      }}
      variant="ghost"
      className="h-[50px] w-[50px] rounded-full p-
0"
    >
      <Image
        src={'/images/google-logo.svg'}
        className="rounded-full"
        width={50}
        height={50}
        style={{
          aspectRatio: '1',
          objectFit: 'cover',
        }}
        alt="LogIn by google"
      />
    </Button>
    <Button
      onClick={async () => {
        await signIn('facebook', { callbackUrl:
'/' });

      }}
      variant="ghost"
      className="h-[50px] w-[50px] rounded-full p-
0"
    >
      <Image
        src={'/images/facebook-logo.svg'}
        className="rounded-full"
        width={50}
        height={50}
        style={{
          aspectRatio: '1',
          objectFit: 'cover',
        }}
        alt="LogIn by facebook"
      />
    </Button>
  </div>
  <p className="text-neutral-400">a6o</p>
  <Tabs defaultValue="email" className="w-2/3">
    <TabsList className="flex h-14 w-full rounded-b-
none p-0">
      <TabsTrigger

```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        className="m-2 grow py-2 data-
[state=active]:mb-0 data-[state=active]:rounded-t-md data-
[state=active]:py-3 data-[state=active]:shadow-none"
        value="email"
    >
        Електронна пошта
    </TabsTrigger>
    <TabsTrigger
        value="account"
        className="m-2 grow py-2 data-
[state=active]:mb-0 data-[state=active]:rounded-t-md data-
[state=active]:py-3 data-[state=active]:shadow-none"
    >
        Облікові дані
    </TabsTrigger>
</TabsList>
<TabsContent
    value="account"
    className="mt-0 rounded-b-md bg-neutral-100
p-2 pt-0"
>
    <form
        method="post"
        action="/api/auth/callback/credentials"
        className="flex h-52 flex-col justify-
center gap-4 rounded-b-md rounded-tl-md bg-white p-4"
    >
        <Input
            name="csrfToken"
            type="hidden"
            defaultValue={csrfToken}
        />
        <Input
            name="username"
            type="text"
            placeholder="Введіть ваш логін..."
            className="border border-neutral-400"
        />
        <Input
            name="password"
            type="password"
            placeholder="Введіть ваш пароль..."
            className="border border-neutral-400"
        />
        <Button className="w-full" type="submit">
            Увійти
        </Button>
    </form>
</TabsContent>
<TabsContent
    value="email"

```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        className="mt-0 rounded-b-md bg-neutral-100
p-2 pt-0"
        >
        <form
            method="post"
            action="/api/auth/signin/email"
            className="flex h-52 flex-col justify-
center gap-4 rounded-b-md rounded-tr-md bg-white p-4"
        >
            <Input
                name="csrfToken"
                type="hidden"
                defaultValue={csrfToken}
            />
            <Input
                name="email"
                type="email"
                placeholder="Введіть вашу пошту..."
                className="border border-neutral-400"
            />
            <Button className="w-full" type="submit">
                Увійти
            </Button>
        </form>
    </TabsContent>
</Tabs>
</div>
</div>
<style>
    {`
        html {
            overflow-y: auto;
        }
    `}
</style>
</>
);
};

```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

Donate page

```

const Donate: React.FC = () => {
  const router = useRouter();

  const [isDialogOpen, setIsDialogOpen] = useState(false);

  const [isSubscription, setIsSubscription] = useState(false);

  function changeDialogState() {
    setIsDialogOpen(!isDialogOpen);
  }

  const form = useForm<z.infer<typeof UserCNBSchema>>({
    resolver: zodResolver(UserCNBSchema),
    defaultValues: {
      action: PaymentAction.PAYDONATE,
      amount: 100,
      currency: PaymentCurrency.UAH,
      subscribe_periodicity: LiqpayPeriodicity.MONTH,
    },
  }));

  function handleSubscription() {
    if (form.getValues('action') === PaymentAction.SUBSCRIBE) {
      setIsSubscription(true);
    } else {
      setIsSubscription(false);
    }
  }

  async function onSubmit(values: z.infer<typeof UserCNBSchema>) {
    const params = new URLSearchParams({
      ...values,
      amount: values.amount.toString(),
    });

    const res = await axios.get(`/api/payment/generate`, {
      params,
    });

    if (res.status === 200 && typeof res.data === 'string') {
      await router.push(res.data);
    }
  }

  return (
    <Layout className="items-center bg-homepage-cover py-14">
      <Head>
        <title>Пітримати</title>
      </Head>

```

		Чернюк А.М.			ІПЗ.КР.Б – 121 – 24 – ІЗ	Арк.
		Савицький Р.С.				78
Змн.	Арк.	№ докум.	Підпис	Дата		


```

Будемо раді вашій підтримці! Кожна ваша гривня
робити світ кращим місцем, де кожен може знайти
розуміння.
</p>
<Button
  className="mt-3 self-center"
  onClick={changeDialogState}
>
  Підтримати зараз
</Button>
</div>
<Dialog open={isDialogOpen} onOpenChange={setIsDialogOpen}>
  <DialogContent className="flex flex-col">
    <DialogHeader className="self-start">
      <DialogTitle className="text-xl">
        Введіть дані для оплати
      </DialogTitle>
    </DialogHeader>
    <Form {...form}>
      <form
        onSubmit={form.handleSubmit(onSubmit)}
        className="flex flex-col gap-4 py-4"
      >
        <FormField
          control={form.control}
          name="action"
          render={({ field }) => (
            <FormItem className="flex flex-col
              gap-y-2">
              <div className="flex w-full
                justify-between">
                <FormLabel>
                  Введіть дані для оплати
                </FormLabel>
                <FormControl>
                  <ToggleGroup
                    type="single"
                    className="flex-grow"
                    onValueChange={({
                      e: PaymentAction,
                    }) => {
                      if (e) {
                        setPaymentAction(e);
                      }
                    }}
                    {...field}
                  >
                    <ToggleGroupItem

```

		Чернюк А.М..			ІПЗ.КР.Б – 121 – 24 – ІЗ	Арк.
		Савіцький Р.С.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

```

PaymentAction.PAYDONATE
value={
}
className="flex-
grow rounded-none rounded-s-lg border border-e-0 data-[state=on]:border-e
data-[state=on]:border-black"
aria-
label="PAYDONATE"
>
    Разовий платіж
</ToggleGroupItem>
<ToggleGroupItem
    value={
PaymentAction.SUBSCRIBE
}
className="flex-
grow rounded-none rounded-e-lg border border-s-0 data-[state=on]:border-s
data-[state=on]:border-black"
aria-
label="SUBSCRIBE"
>
    Підписка
</ToggleGroupItem>
</ToggleGroup>
</FormControl>
</div>
<FormMessage className="text-
center" />
</FormItem>
)}
/>
{isSubscription && (
    <FormField
        control={form.control}
        name="subscribe_periodicity"
        render={({ field }) => (
            <FormItem className="space-y-2">
                <FormLabel className="text-
lg">
                    Регулярність платежу:
                </FormLabel>
                <FormControl>
                    <ToggleGroup
                        type="single"
                        className="flex-grow
gap-0 rounded-lg"
                        onValueChange={

```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        e:
LiqpayPeriodicity,
    ) => {
        if (e) {

        }
    }}
    {...field}
>
    <ToggleGroupItem
        value={

LiqpayPeriodicity.DAY
        }
        className="flex-
grow rounded-none rounded-s-lg border border-e-0 data-[state=on]:border-e
data-[state=on]:border-black"
        aria-label="DAY"
    >
        Щодня
    </ToggleGroupItem>
    <ToggleGroupItem
        value={

LiqpayPeriodicity.MONTH
        }
        className="flex-
grow rounded-none border data-[state=on]:border-black"
        aria-
label="MONTH"
    >
        Щомісяця
    </ToggleGroupItem>
    <ToggleGroupItem
        value={

LiqpayPeriodicity.YEAR
        }
        className="flex-
grow rounded-none rounded-e-lg border border-s-0 data-[state=on]:border-s
data-[state=on]:border-black"
        aria-label="YEAR"
    >
        Щороку
    </ToggleGroupItem>
    </ToggleGroup>
    </FormControl>
    <FormMessage />
    </FormItem>
    )}

```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    />
  })
  <FormField
    control={form.control}
    name="currency"
    render={({ field }) => (
      <FormItem className="space-y-2">
        <FormLabel className="text-lg">
          Валюта:
        </FormLabel>
        <FormControl className="w-full">
          <ToggleGroup
            type="single"
            className="flex-grow gap-
0 rounded-lg"
            onChange={
              e: PaymentCurrency,
            } => {
              if (e) {
                field.onChange(e);
              }
            }
          <ToggleGroupItem
            value={PaymentCurrency.UAH}
            className="flex-grow
rounded-none rounded-s-lg border border-e-0 data-[state=on]:border-e
data-[state=on]:border-black"
            aria-label="UAH"
          >
            ₴
          </ToggleGroupItem>
          <ToggleGroupItem
            value={PaymentCurrency.USD}
            className="flex-grow
rounded-none border data-[state=on]:border-black"
            aria-label="USD"
          >
            $
          </ToggleGroupItem>
          <ToggleGroupItem
            value={PaymentCurrency.EUR}
            className="flex-grow
rounded-none rounded-e-lg border border-s-0 data-[state=on]:border-s
data-[state=on]:border-black"
            aria-label="EUR"
          >
            €
          </ToggleGroupItem>
        </FormControl>
      </FormItem>
    )}
  </FormField>

```

		Чернюк А.М.			ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
		Савіцький Р.С.				83
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        >
        €
        </ToggleGroupItem>
    </ToggleGroup>
</FormControl>
<FormMessage />
</FormItem>
    })
/>
<FormField
    control={form.control}
    name="amount"
    render={({ field }) => (
        <FormItem className="space-y-2">
            <FormLabel className="text-lg">
                Сума внеску:
            </FormLabel>
            <FormControl className="flex-
grow">
                <Input type="number"
                />
            </FormControl>
            <FormMessage />
        </FormItem>
    )}
/>
<DialogFooter className="self-end">
    <Button type="submit">Оплатити</Button>
</DialogFooter>
</form>
</Form>
</DialogContent>
</Dialog>
</Layout>
    );
};

```