

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 «Вебтехнології»)

на тему:

«Додаток ментального здоров'я «MindPlace»»

Виконала студентка групи ВТ-21-1
ПШЦУК Софія Олександрівна

Керівник роботи:
ЧИЖМОТРИЯ Олексій Володимирович

Рецензент:
САВІЦЬКИЙ Роман Святославович

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

«14» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ПІЩУК Софія Олександрівна**

Керівник роботи: **ЧИЖМОТРЯ Олексій Володимирович**

Тема роботи: **«Додаток ментального здоров'я «MindPlace»»,**

затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**

Вихідні дані для роботи: **методи та засоби моделювання, проєктування та реалізації програмного забезпечення для додатку ментального здоров'я «MindPlace» за допомогою сучасних інформаційних технологій для автоматизації взаємодії між користувачами, контентом та модераторами.**

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Чижмотря О.В.	17.02.2025	17.02.2025
2	Чижмотря О.В.	19.02.2025	19.02.2025
3	Чижмотря О.В.	21.02.2025	21.02.2025

РЕФЕРАТ

Пояснювальна записка до випускної кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновку та списку використаних джерел.

Текстова частина викладена на 69 сторінках друкованого тексту. Список використаних джерел містить 30 найменувань і займає 3 сторінки. Пояснювальна записка має 22 сторінки додатків. В роботі наведено 56 рисунків та 21 таблиць. Загальний обсяг роботи – 92 сторінки.

У роботі поставлено основні завдання щодо розробки вебзастосунку для ментального здоров'я «MindPlace». Система включає модулі для аутентифікації користувачів, вибору ролі (пацієнт або спеціаліст), ведення особистого кабінету, отримання та надання консультацій, публікації та модерації блогових статей, а також функції конфіденційності та безпеки даних. Реалізовано інтеграції з Google Calendar та Google Meet, а також авторизацію через Google. Застосунок побудовано на базі архітектури MVC із використанням сучасних вебтехнологій: PHP, фреймворку Yii2, системи керування базами даних MySQL та бібліотеки інтерфейсу Bootstrap 5. Забезпечено підтримку безпечного зберігання медичних документів. Особливу увагу приділено UX-дизайну та модульності системи для подальшого масштабування.

Ключові слова: ПЛАТФОРМА, КОРИСТУВАЧ, ДОСТУП, БАЗА ДАНИХ, МОДЕРАЦІЯ.

					ІПЗ.КР.Б – 121 –25 – ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Піщук С.О.			Додаток ментального здоров'я «MindPlace» Пояснювальна записка	Літ.	Арк.	Аркушів	
Керівник		Чижмоторя О.В.					4	94	
Рецензент		Савіцький Р.С.							
Зав. каф.		Вакалюк Т.А.							
						Житомирська політехніка, Група ВТ-21-1			

ABSTRACT

The explanatory note to the bachelor's thesis consists of an introduction, three chapters, a conclusion and a list of references.

The text part is set out on 69 pages of printed text. The list of references contains 30 titles and occupies 3 pages. The explanatory note includes 22 pages of appendices. The paper contains 56 figures and 21 tables. The total volume of the paper is 92 pages.

The paper sets out the main tasks of developing a web-based application for mental health 'MindPlace'. The system includes modules for user authentication, role selection (patient or specialist), personal account management, receiving and providing consultations, publishing and moderating blog articles, as well as data privacy and security functions. Integrations with Google Calendar and Google Meet, as well as Google authorisation have been implemented. The application is built on the MVC architecture with the use of modern web technologies: PHP, Yii2 framework, MySQL database management system, and Bootstrap 5 interface library. Support for secure storage of medical documents is provided. Particular attention was paid to the UX design and modularity of the system for further scaling.

Keywords: PLATFORM, USER, ACCESS, DATABASE, MODERATION.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ, ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ	9
1.1. Визначення та постановка задачі	9
1.2. Аналіз існуючих аналогів програмного продукту	10
1.3. Визначення архітектури ПЗ	14
1.4. Обґрунтування вибору інструментальних засобів та вимоги до ПЗ.....	16
Висновки до першого розділу	18
РОЗДІЛ 2. ПРОЄКТУВАННЯ ДОДАТКУ МЕНТАЛЬНОГО ЗДОРОВ'Я «MINDPLACE».....	19
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи	19
2.2. Розробка бази даних системи.....	25
2.3. Проєктування та реалізація алгоритмів роботи платформи	30
2.4. Реалізація функціоналу додатку	33
Висновки до другого розділу	39
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ДОДАТКОМ МЕНТАЛЬНОГО ЗДОРОВ'Я «MINDPLACE».....	41
3.1. Порядок встановлення та налаштування параметрів системи	41
3.2. Структура інтерфейсу додатку ментального здоров'я.....	42
3.3. Тестування роботи програмного комплексу	61
Висновки до третього розділу.....	63
ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67
Додатки.....	70

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

PHP – Hypertext Preprocessor

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

ООП – Об’єктно-орієнтовне програмування

ПЗ – Програмне забезпечення

БД – База даних

MVC – Model-view-controller

CRUD – Create, Read, Update, Delete

Amazon S3 – Amazon Simple Storage Service

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

У пояснювальній записці до випускної кваліфікаційної роботи бакалавра буде наведено процес розробки платформи для отримання та надання кваліфікованої психологічної допомоги. Система орієнтована на автоматизацію взаємодії між користувачами, контентом та модераторами, а також має забезпечити надійну і швидку роботу для великої кількості користувачів.

Актуальність теми зумовлена тим, що у сучасному світі все більше людей стикаються з проблемами ментального здоров'я, особливо на фоні війни, які можуть негативно впливати на якість їхнього життя, роботу, навчання та стосунки. Водночас, стигматизація теми психічного здоров'я часто ускладнює пошук кваліфікованої допомоги. Тому створення вебсайту, де користувачі можуть отримати доступ до перевірених фахівців у сфері психології, а також читати корисні матеріали про ментальне здоров'я, є актуальним і необхідним. Це дозволить людям отримувати своєчасну допомогу та розширювати свої знання про збереження психологічного благополуччя.

Метою роботи є проєктування та реалізація платформи для надання та отримання кваліфікованої психологічної допомоги та поширення перевіреної інформації про ментальне здоров'я, що забезпечить швидку взаємодію між користувачами, ефективні інструменти комунікації та якісну систему модерації. Реалізація цієї системи дозволить забезпечити стабільну роботу при великому навантаженні, підвищити комфортність використання та сприяти розвитку обізнаності про важливість ментального здоров'я.

Завданням випускної кваліфікаційної роботи бакалавра є виконання таких задач:

- постановка основних задач системи та узагальнення її структури;
- аналіз аналогів;
- побудова архітектури, що буде задовольняти всі вимоги системи;
- вибір та обґрунтування засобів реалізації системи.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Об’єкт дослідження: інформаційні технології для автоматизації взаємодії між користувачами, контентом та модераторами.

Предмет дослідження: методи та засоби моделювання, проєктування та реалізації програмного забезпечення для додатку ментального здоров’я «MindPlace».

.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ, ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1. Визначення та постановка задачі

Основною метою розробки додатку є реалізувати легке надання та отримання психологічних послуг та отримання інформації про ментальне здоров'я для звичайних користувачів. MindPlace слугує централізованою платформою, яка сприяє ефективній комунікації, співпраці між спеціалістами і користувачами, доступу до інформаційних ресурсів про ментальне здоров'я. Вона має на меті забезпечити гнучке, доступне та безпечне середовище, спростити адміністративні завдання.

У випускній кваліфікаційній роботі необхідно розробити вебсистему, яка буде містити такий функціонал:

Функції клієнтської частини для користувача:

- перегляд власного профілю, редагування певних полів;
- фільтрація та пошук психологів та психотерапевтів з фільтрацією;
- запис на консультацію в онлайн форматі;
- отримання консультацій;
- написання відгуків про спеціалістів;
- перегляд історії звернень;
- перегляд доступних статей про психологічне здоров'я.

Функції клієнтської частини для спеціаліста:

- перегляд власного профілю, редагування певних полів;
- завантаження документів та сертифікатів для підтвердження своєї кваліфікації;
- перегляд заявок клієнтів;
- керування своїм розкладом;
- зв'язок з клієнтами через Google Meet [1];
- додаткове збереження історії сесій (подій) в Google Calendar [2];
- створення медичних інформаційних статей.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Функції для адміністраторів:

- здійснення модерації відображення статей;
- створення, редагування та видалення класів;
- керування ролей користувачів.

Результатом реалізації поставленого завдання є функціональна платформа для отримання психологічної допомоги та інформації про ментальне здоров'я.

1.2. Аналіз існуючих аналогів програмного продукту

Розглянемо наявні аналоги, які найбільше подібні до теми кваліфікаційної роботи: Mentoly, Розкажи мені, Mindly, наведемо їх можливості та порівняємо їх. Ці платформи служать безпечним місцем для людей, які прагнуть отримати психологічну допомогу, налагодити свій душевний стан та дізнатися більше про ментальне здоров'я. Вивчаючи ці платформи, ми прагнемо зрозуміти загальні особливості та характеристики, які роблять їх популярними в наданні допомоги в онлайн режимі.

Mentoly [3] – це популярний вебсайт з надання психологічної допомоги та інформації про моральне здоров'я (Рис. 1.1). Тут доступні такі функції як персоналізований підбір психолога, запис та отримання консультацій в онлайн режимі чи наживо, блог на цю тематику та просто поради щодо переживання складних моментів у житті. Цим додатком користується багато людей та кваліфікованих психологів.

Отже серед переваг можна виокремити такі:

- інтеграція акаунту з фейсбуком та гугл акаунтом
- легкість використання та зрозумілий інтерфейс навіть для тих хто відвідує цей сайт вперше
- надання звітності спеціалістам
- на сторінці спеціаліста присутня інформація про нього та його методи роботи, освіту та часом відео з коротким ознайомленням зроблене саме цим лікарем

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Натомість серед недоліків можемо виокремити платність послуг. Це робить доступність допомоги трохи меншою.

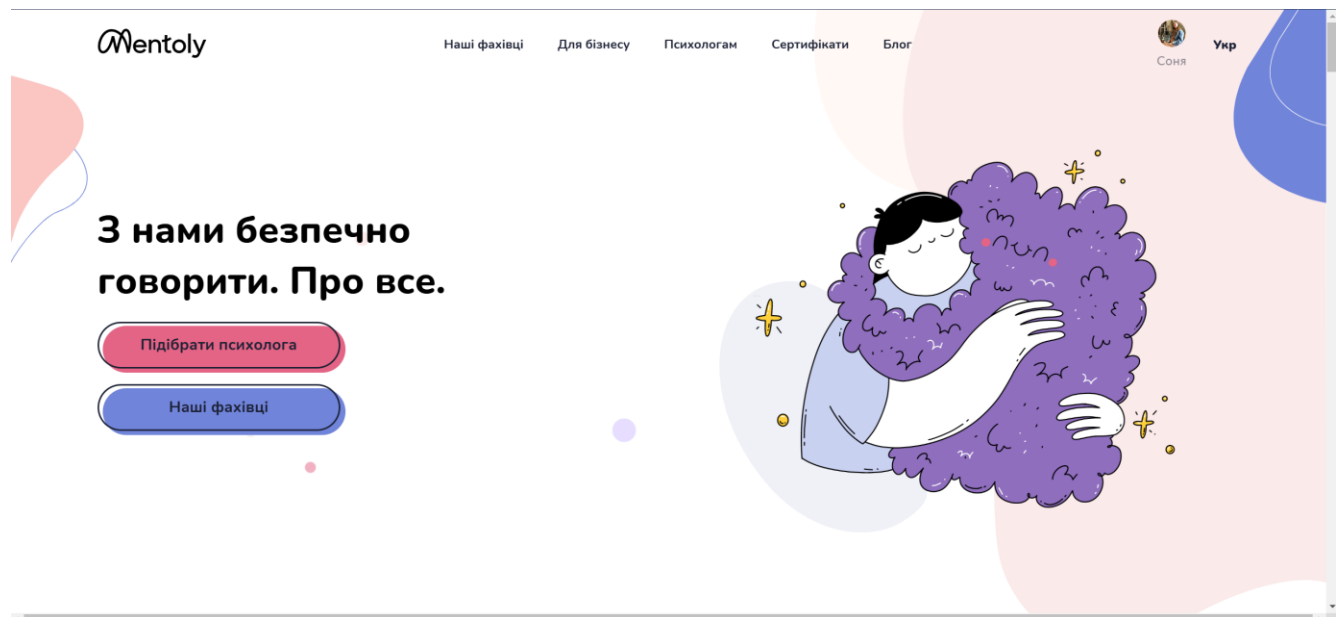


Рис. 1.1. Основна сторінка вебсайту Mentoly

«Розкажи мені» [4] – це безкоштовна інтернет-платформа для психологічних консультацій (Рис. 1.2). Команда платформи працює здебільшого за методом когнітивно-поведінкової терапії (КПТ), яка є однією з найпоширеніших форм психотерапевтичної допомоги людям з різними психологічними проблемами.

Тут працюють лише кваліфіковані спеціалісти, з якими клієнт може відверто обговорити ситуації, що його турбують. Платформа розроблена та створена Інститутом глобальної трансформації (Institute for Global Transformation).

Переваги:

- допомога від спеціалістів є безкоштовною
- на сайті працює понад 1000 лікарів
- замість блогу є посилання на їхній телеграм канал з корисною інформацією та порадами щодо побутових і кризових ситуацій
- інтерфейс сайту є легким у використанні

Недоліки:

- людина не може самостійно обрати собі фахівця, його підбирають адміністратори платформи в залежності від запиту

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

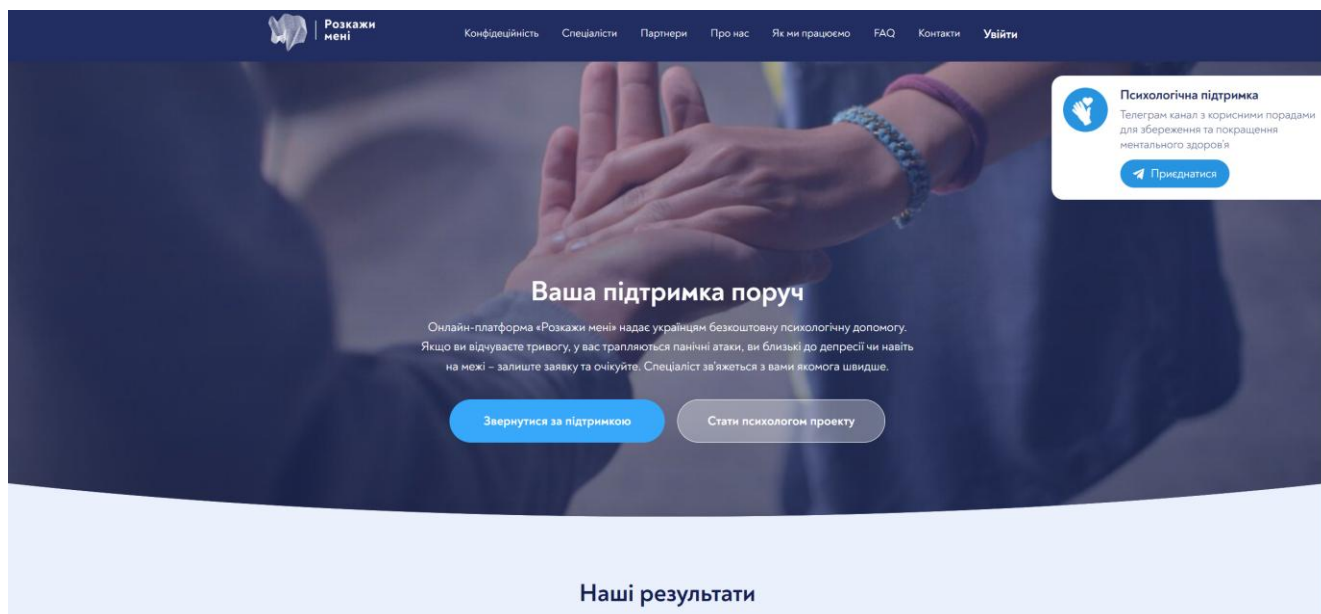


Рис. 1.2. Основна сторінка вебсайту «Розкажи мені»

Mindly [5] – це платформа з надання платної психологічної та психіатричної допомоги (Рис. 1.3). Для початку користувач проходить анкету підбору психологів вказуючи свої дані психоемоційного стану на цей момент, вік та вподобання щодо вибору і методів фахівців. Тут доступні спеціалісти для безлічі різних запитів, а також вони проходять відбір аби працювати саме тут.

З переваг:

- є мобільний додаток аби звернення по допомогу можна було здійснити з будь-якого зручного місця та навіть не маючи ПК
- з рекомендованих фахівців можливо обрати того чия анкета найбільше підходить до запиту
- на сторінці спеціаліста присутня інформація про нього та його методи роботи, освіту та часом відео з коротким ознайомленням зроблене саме цим лікарем

Недоліки:

- зручність інтерфейсу було розроблено тільки під мобільний додаток
- послуги є платними і з доволі високими цінами за одну консультацію
- щодо підбору спеціаліста відбувається незрозуміла ситуація, адже було помічено, що деякі виставлені користувачем критерії ігноруються

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

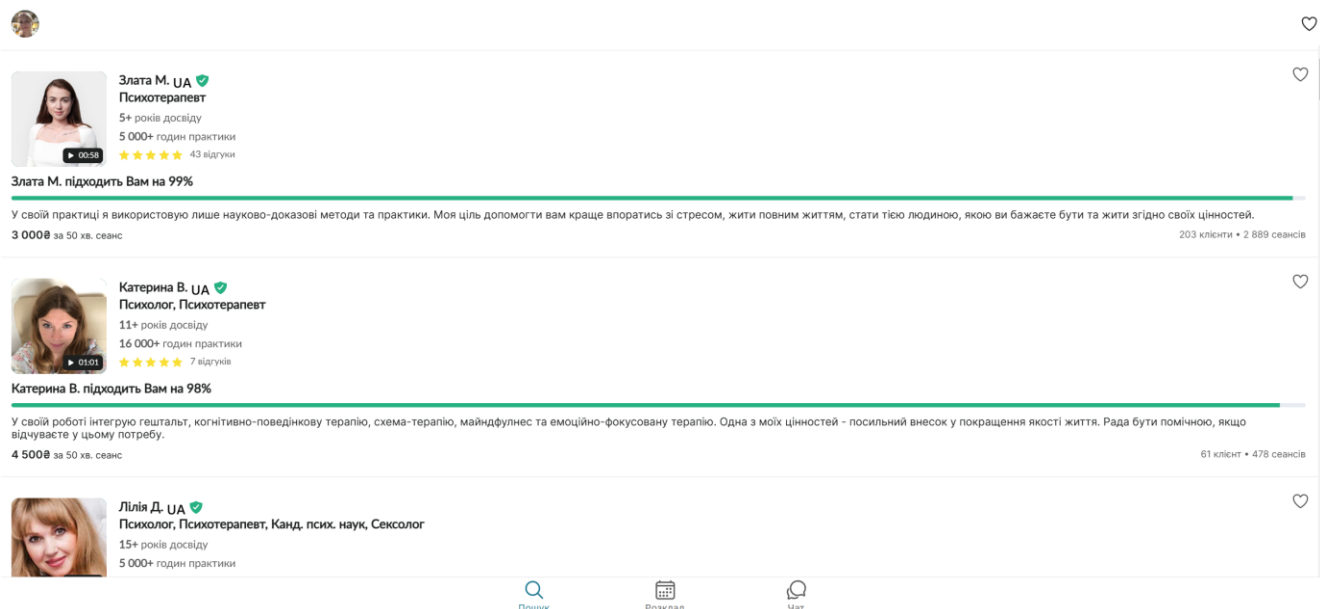


Рис. 1.3. Основна сторінка вебсайту Mindly

Отже аби підсумувати усю зазначену вище інформацію складемо таблицю порівняння основних критеріїв обраних платформ.

Таблиця 1.1

Порівняльна таблиця наявних рішень на ринку

Критерій	Mentoly	«Розкажи мені»	Mindly	MindPlace
Доступність	Вебсайт	Вебсайт	Вебсайт (у вигляді моб. додатку), мобільний додаток	Вебсайт
Формати взаємодії	Відео, аудіо, чат, особиста консультація	Чат	Відео, аудіо, чат	Відео зв'язок (Google Meet)
Реєстрація користувачів	Так	Так	Так	Так
Вибір психолога	Так	Ні	Так	Так
Алгоритм підбору спеціаліста	Так	Ні	Так	Так
Вартість послуг	Платно	Безкоштовно	Платно	Безкоштовно
Конфіденційність	Анонімність, захист даних	Захист даних	Захист даних	Захист даних
Додаткові ресурси	Блог, подарункові сертифікати	Посилання на сторонні додатки з порадами	Немає	Блог

Розглянувши та проаналізувавши представлені вище існуючі аналоги ми можемо сформулювати необхідні характеристики для створення сайту психологічної допомоги та ментального здоров'я:

1. Доступність та зручність використання, що включає в себе адаптивний дизайн для різних пристроїв та інтуїтивно зрозумілий інтерфейс.
2. Основний функціонал для користувачів: забезпечення реєстрації та авторизації, власний профіль, фільтрація доступних лікарів, онлайн-календар запису на сеанс, отримання консультації через відео-дзвінок та система відгуків та рейтингів.
3. Функціонал для психологів також включатиме можливість керування власним розкладом, доступ до інформації про клієнта після його запису на сеанс та інструменти для проведення онлайн-сеансів, форми написання та надсилання запиту на опублікування медичних статей.
4. Безпека та конфіденційність, що потребує шифрування даних, захист персональної інформації та політику конфіденційності.
5. Додаткові можливості, такі як блог з корисними матеріалами про ментальне здоров'я та підтримка мультимовності.

Таким чином, аналіз існуючого ПЗ дає змогу визначити функціональні особливості та технічні вимоги, які варто врахувати під час розробки власного додатку. Метою є створення продукту, який задовольнятиме потреби користувачів та сприятиме ефективному наданню допомоги та інформації людям які цього потребують.

1.3. Визначення архітектури ПЗ

Для розробки вебсайту ментального здоров'я "MindPlace" було обрано архітектурний шаблон MVC [6]. Використання цієї архітектури забезпечить чітке розділення логіки програми, що сприятиме зручному розширенню, підтримці та повторному використанню компонентів.

Обрана архітектура складається з трьох основних компонентів: моделі, представлення та контролера.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Модель відповідає за управління всіма даними в системі — від інформації про користувачів і психологів до записів на сесії. Саме в ній реалізована бізнес-логіка, зокрема алгоритми підбору відповідного спеціаліста. Уся інформація зберігається в базі даних, включаючи історію консультацій.

Представлення — це візуальна частина програми, з якою взаємодіє користувач. Інтерфейс адаптований для різних пристроїв і включає сторінки для реєстрації, пошуку спеціалістів, бронювання сесій та перегляду історії.

Контролер обробляє дії користувачів, такі як реєстрація, вхід до системи, запис на сесію та перегляд результатів. Він є посередником між моделлю та представленням, забезпечуючи коректну взаємодію між ними. Також контролер підключається до зовнішніх сервісів, наприклад, календарів або google-акаунту.

Обрана архітектура має кілька важливих переваг. Вона модульна, тобто дозволяє вносити зміни або оновлення в одну частину системи без шкоди для інших. Вона гнучка — це дає змогу легко додавати нові функції, такі як чат-боти або рекомендаційні системи. Масштабованість дозволяє системі ефективно працювати навіть із великою кількістю користувачів. А централізоване управління доступом до даних забезпечує високий рівень безпеки, мінімізуючи ризики витоку конфіденційної інформації.

Розглянемо цикл обробки запиту, оскільки він є ключовим елементом клієнтсько-серверної архітектури. Після отримання запиту від користувача сервер ініціює виконання функції, яка обробляє запити до бази даних та управляє даними сесії. Далі перевіряється рівень доступу користувача до запитуваної сторінки. Якщо доступ відсутній, система перенаправляє його на сторінку з повідомленням про помилку. У разі підтвердження доступу запит обробляється: за необхідності сервер звертається до бази даних і здійснює обробку отриманих даних. Після цього формується відповідне представлення, і згенерований HTML-код надсилається користувачеві. У межах розробки передбачено, що взаємодія користувача з сервером відбуватиметься виключно через графічний інтерфейс, що узгоджується з концепцією MVC (Рис. 1.4).

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

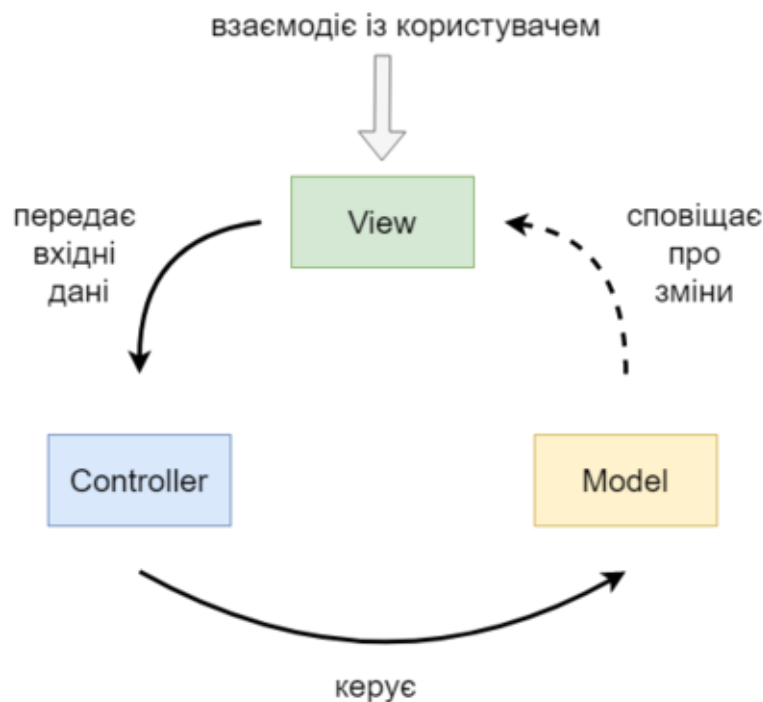


Рис. 1.4. Загальний принцип роботи патерну MVC

Обрана MVC-архітектура дозволить вебсайту "MindPlace" забезпечити високу продуктивність, зручність використання та надійність, що є ключовими аспектами для платформ психологічної підтримки.

1.4. Обґрунтування вибору інструментальних засобів та вимоги до ПЗ

У цьому проєкті для створення локального середовища розробки було обрано Docker [7], що є сучасним та ефективним інструментом для контейнеризації застосунків. Його використання дозволяє створити ізольоване середовище з усіма необхідними сервісами, незалежно від операційної системи розробника. Docker забезпечує відтворюваність середовища, що особливо важливо для командної розробки, автоматизації CI/CD-процесів та усунення конфліктів, пов'язаних із відмінностями між локальним і продакшн-середовищем. Для створення самого вебзастосунку було обрано PHP-фреймворк Yii2 [8] у поєднанні з реляційною системою керування базами даних MySQL [9].

Вибір Yii2 обумовлений його високою продуктивністю, розвиненим функціоналом і підтримкою архітектурного підходу MVC, що забезпечує поділ логіки, представлення й доступу до даних. Завдяки вбудованим засобам, таким як ORM ActiveRecord, генератор коду Gii, система керування доступом RBAC,

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

підтримка REST API та кешування, Yii2 дозволяє суттєво прискорити розробку складних систем [12]. Крім того, його добре структурована документація та активна спільнота роблять фреймворк надійною основою для масштабованих вебпроектів.

Вибір MySQL як системи керування базами даних зумовлений її стабільністю, високою швидкістю обробки запитів та широкою підтримкою у веброзробці. MySQL є відкритим програмним забезпеченням із великою спільнотою, що дозволяє швидко знаходити рішення типових проблем і отримувати оновлення без додаткових витрат. Вона повністю сумісна з Yii2 через ORM ActiveRecord, що дає змогу зручно працювати з таблицями бази даних у вигляді об'єктів. MySQL також забезпечує надійність зберігання даних, підтримку транзакцій, індексацію та ефективну роботу з великими обсягами інформації, що робить її оптимальним вибором для розробки функціональної та масштабованої платформи, орієнтованої на реальне навантаження.

Для створення інтерфейсу користувача використано HTML5 та CSS3 [13], що забезпечують семантичну розмітку й гнучке стилізування. З метою пришвидшення розробки, забезпечення адаптивності та візуальної цілісності дизайну було використано фреймворк Bootstrap 5 [14], а також власні стилі, реалізовані з використанням препроцесора SASS [15], що полегшує підтримку та повторне використання CSS-коду.

Клієнтська частина застосунку розроблена з урахуванням кросбраузерної сумісності – вона стабільно працює в популярних браузерах, таких як Google Chrome, Mozilla Firefox та Opera. Окрему увагу приділено адаптивності інтерфейсу, щоб забезпечити коректне відображення та зручну навігацію як на десктопних, так і на мобільних пристроях, незалежно від розміру екрана.

Загалом, обрані інструменти та технології дозволили побудувати сучасну, надійну та масштабовану вебплатформу, що відповідає актуальним вимогам до зручності, стабільності та розширюваності. Такий технічний стек забезпечує ефективну роботу як на етапі розробки, так і під час подальшої експлуатації системи, спрощуючи інтеграцію нових модулів, оновлення й підтримку.

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

Окрім базових технологій, система підтримує інтеграцію зі сторонніми сервісами, такими як Google Meet та Google Calendar, що дає змогу реалізувати онлайн-консультації та автоматизоване додавання подій у календар спеціаліста. Також реалізовано завантаження й зберігання документів у хмарному сховищі Amazon S3, що забезпечує безпечний і масштабований спосіб роботи з файлами. Така інтеграція значно розширює функціональність проєкту та підвищує його практичну цінність для користувачів.

Висновки до першого розділу

Під час підготовки першого розділу було здійснено ґрунтовний аналіз предметної області, що дозволило визначити ключові аспекти майбутнього додатка для підтримки ментального здоров'я.

У процесі вивчення існуючих аналогів було прийнято рішення забезпечити зручний і доступний функціонал для користувачів, а також створити унікальний та інтуїтивно зрозумілий інтерфейс.

Для розробки системи було обрано архітектурний шаблон MVC, а також PHP-фреймворк Yii2, що обґрунтовано його перевагами. Було складено перелік сучасних інструментів для веброботи, а для зберігання даних обрана реляційна СУБД MySQL.

На стороні клієнта для створення інтерфейсу та стилізації застосовуються стандартні вебтехнології, зокрема HTML5 і CSS із використанням бібліотеки Bootstrap 5. Окрему увагу приділено сумісності з різними веббраузерами та платформами, що забезпечує зручний і стабільний доступ до додатка.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ДОДАТКУ МЕНТАЛЬНОГО ЗДОРОВ'Я «MINDPLACE»

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Система «MindPlace» є вебплатформою для підтримки ментального здоров'я шляхом надання доступу до перевірених спеціалістів, освітнього контенту та зручних інструментів комунікації. Вона побудована за патерном MVC з використанням фреймворку Yii2 для серверної частини, MySQL для управління базою даних, а також Bootstrap 5 для інтерфейсної частини. Основна логіка системи включає реєстрацію та вхід користувачів, вибір ролі (звичайний користувач або спеціаліст із перевіркою документів), ведення особистого кабінету, можливість планування онлайн-консультацій, проведення цих консультацій онлайн у відео-режимі, а також доступ до блогу з медичними статтями.

Платформа підтримує інтеграцію з Google Calendar для створення події і нагадування та Google Meet для проведення консультації в онлайн режимі. Також авторизацію через Google із забезпеченням конфіденційності користування. Усі дані користувачів зберігаються із дотриманням сучасних вимог до безпеки.

Відповідно до вимог було сформовано варіанти використання системи. У системі «MindPlace» розрізняють три ключові ролі: адміністратор, спеціаліст та пацієнт.

Користувач може редагувати свій профіль, переглядати статті та спеціалістів, спілкуватися зі спеціалістами та залишати про них відгуки. Спеціаліст (лікар) створює та редагує власні статті, редагує профіль, проводить консультації з користувачами. Адміністратор має доступ до редагування доступу та ролей користувачів та модерації статей. Для наглядності було створено діаграми онлайн [16]. Демонстрація діаграмою показана на рисунку 2.1.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

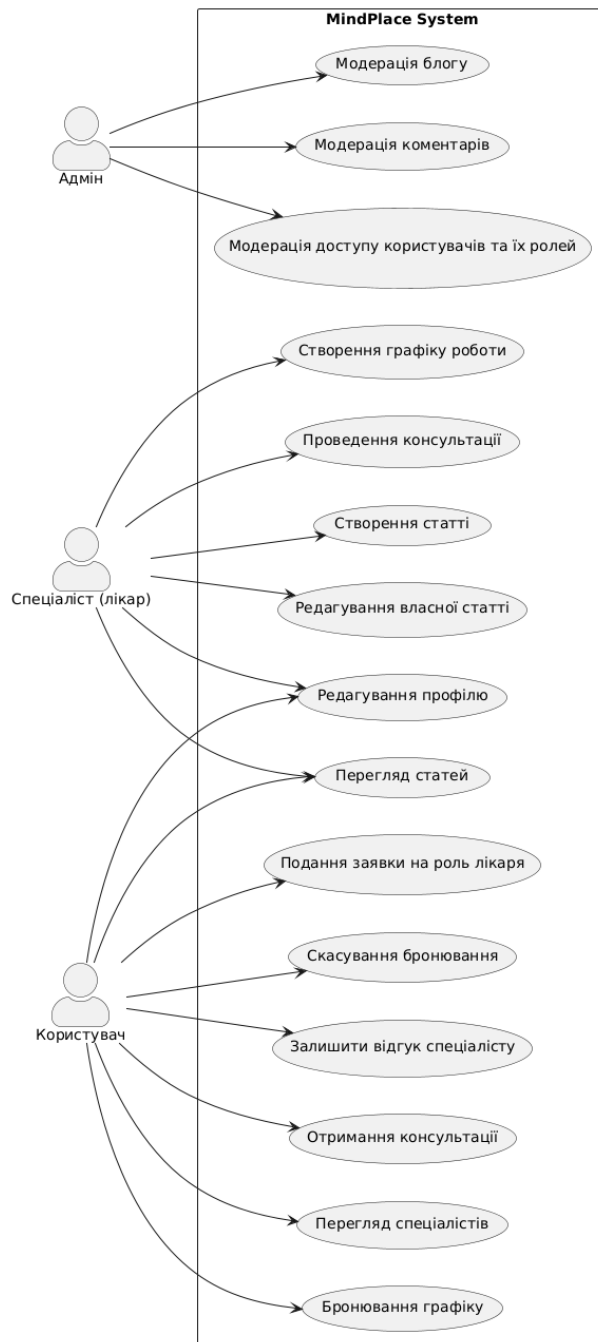


Рис. 2.1. Діаграма варіантів використання
Основні прецеденти описані в таблицях 2.1 - 2.14.

Таблиця 2.1

Опис прецеденту «Редагування профілю»

Назва	Редагування профілю
Опис	Користувач редагує особисту інформацію у своєму профілі (ім'я, email, пароль тощо). Для спеціалістів можливості розширені до редагування своїх професійних даних.
Виконавці	User, Specialist

Таблиця 2.2

Опис прецеденту «Подання заявки на роль спеціаліста»

Назва	Подання заявки на роль спеціаліста
Опис	Користувач реєструється та надає запрошену інформацію на формі сайту, після чого його заявка стоїть на розгляді адміністраторів.
Виконавці	User

Таблиця 2.3

Опис прецеденту «Керування графіком»

Назва	Керування графіком
Опис	Лікар може створювати та видаляти свої вільні слоти часу роботи.
Виконавці	Specialist

Таблиця 2.4

Опис прецеденту «Перегляд блогу»

Назва	Перегляд блогу
Опис	Користувач переглядає опубліковані статті, написані спеціалістами.
Виконавці	User

Таблиця 2.5

Опис прецеденту «Перегляд спеціалістів»

Назва	Перегляд спеціалістів
Опис	Користувач переглядає профілі доступних спеціалістів (психологів, психіатрів тощо).
Виконавці	User

Таблиця 2.6

Опис прецеденту «Запис на консультацію»

Назва	Запис на консультацію
Опис	Бронювання користувачем слоту часу графіку виставленого спеціалістом для подальшого проведення консультації.
Виконавці	User

Таблиця 2.7

Опис прецеденту «Скасування бронювання»

Назва	Скасування бронювання
Опис	Скасування запису якщо до назначеного часу залишилося не менше години.
Виконавці	User

Таблиця 2.8

Опис прецеденту «Проведення консультації»

Назва	Проведення консультації
Опис	У випадку онлайн-консультації система генерує посилання на приватну кімнату у Google Meet. Якщо консультація очна – пацієнт прибуває за адресом вказаним у лікаря.
Виконавці	User, Specialist

Таблиця 2.9

Опис прецеденту «Створення статей»

Назва	Створення статей
Опис	Спеціаліст створює нову статтю на тему ментального здоров'я.
Виконавці	Specialist

Таблиця 2.10

Опис прецеденту «Редагування своїх статей»

Назва	Редагування своїх статей
Опис	Спеціаліст редагує власні раніше створені статті.
Виконавці	Specialist

Таблиця 2.11

Опис прецеденту «Модерування статей»

Назва	Модерування статей
Опис	Адміністратор переглядає та затверджує/відхиляє створені статті перед публікацією.
Виконавці	Admin

Таблиця 2.12

Опис прецеденту «Модерування відгуків»

Назва	Модерування відгуків
Опис	Адміністратор переглядає й модерує коментарі та відгуки користувачів про спеціалістів.
Виконавці	Admin

Таблиця 2.13

Опис прецеденту «Модерування ролей та доступу користувачів»

Назва	Модерування ролей та доступу користувачів
Опис	Адміністратор змінює ролі користувачів, підтверджує документи спеціалістів, обмежує доступ за порушення.
Виконавці	Admin

Таблиця 2.14

Опис прецеденту «Надсилання відгуків на спеціалістів»

Назва	Надсилання відгуків на спеціалістів
Опис	Користувач залишає відгук про взаємодію зі спеціалістом (текст, рейтинг).
Виконавці	User

Розглянемо діаграму класів [18] на рисунку 2.2. Вона описує структуру користувачів платформи ментального здоров'я, де основною сутністю є клас «User», який реалізує інтерфейс «IdentityInterface». Він містить ключові поля для ідентифікації, авторизації та базової інформації, зокрема ім'я, email, хешований пароль, дата народження, роль, а також токени та інші системні дані. Роль визначає тип користувача: це може бути звичайний користувач, модератор або спеціаліст.

В моделі передбачено створення користувача лише з мінімальними необхідними заповненими даними. Після надсилання заявки на роль лікаря заповнюються усі інші необхідні поля моделі додатковими даними, такими як, місто, стать, контактна інформація, параметри терапії, досвід, освіта, фото, документи та сертифікати. Всі поля, що можуть містити множинні значення, наприклад мови або типи терапії, зберігаються у форматі JSON, що дає гнучкість у

подальшому використанні цих даних. Сама заявка спеціаліста зберігається у моделі «SpecialistApplication» для подальших дій над нею адмінами.

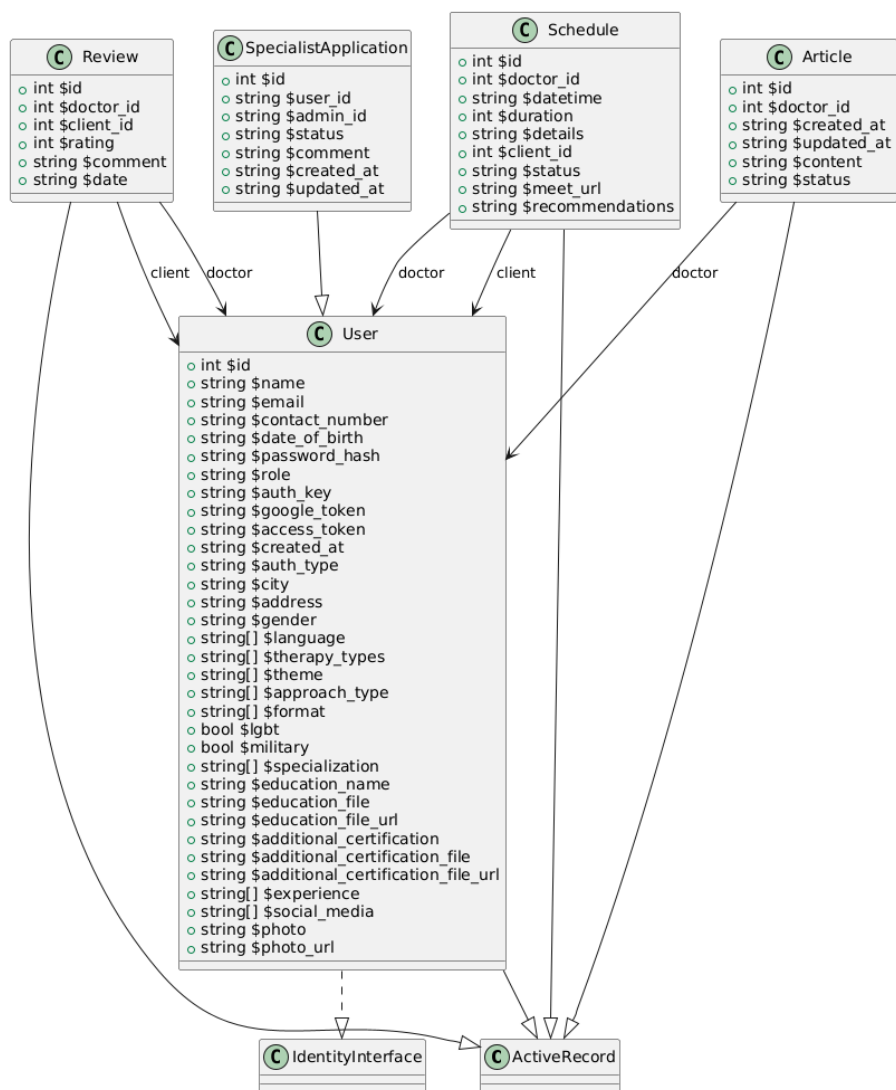


Рис. 2.2. Діаграма класів

Діаграма також враховує інші аспекти платформи, пов'язані зі взаємодією користувачів. Наприклад, лікарі можуть створювати статті, які мають окремі класи «Article» з посиланням на автора. Система дозволяє залишати відгуки про спеціалістів через клас «Review».

Логіка проведення онлайн-консультації здійснюється через модель «Schedule». Новий запис моделі створюється у момент підтвердження спеціалістом слоту графіку консультації. Коли користувач бронює час, дані оновлюються відповідно його додаткових запитів. За годину до назанченого часу, лікар через систему створює активне посилання на Google Meet кімнату, доступ до якої

матимуть обидві сторони та зможуть приєднатися для подальших дій. В процесі бронювання користувач може дозволити створити собі автоматичне нагадування події в Google Callendar через елемент форми.

Уся модель побудована навколо зв'язку один-до-одного між «User» і «SpecialistApplication», та один-до-багатьох між «User» і публікаціями, графіками чи коментарями. Такий підхід дозволяє чітко відокремити базову функціональність від розширених ролей і пов'язаної з ними інформації, зберігаючи при цьому гнучкість і масштабованість архітектури.

2.2. Розробка бази даних системи

Для зберігання усієї критичної інформації системи використовується база даних MySQL. Поля типу string[] зберігаються як JSON у базі (тип text) [19]. Короткий опис можна побачити в таблицях 2.15 - 2.19.

Таблиця 2.15

Таблиця User

Поле	Тип	Nullable
id	integer	ні
name	string	ні
email	string	ні
contact_number	string	так
date_of_birth	string (date)	ні
password_hash	string	ні
role	string	ні
auth_key	string	ні
google_token	string	так
access_token	string	ні
created_at	string (datetime)	ні
auth_type	string	ні
city	string	так
address	string	так
gender	string	так
language	string[]	так
therapy_types	string[]	так

Продовження таблиці 2.15

theme	string[]	так
approach_type	string[]	так
format	string[]	так
lgbt	bool	так
military	bool	так
specialization	string[]	так
education_name	string	так
education_file	string	так
education_file_url	string	так
additional_certification	string	так
additional_certification_file	string	так
additional_certification_file_url	string	так
experience	string[]	так
social_media	string[]	так
photo	string	так
photo_url	string	так

Таблиця 2.16

Таблиця SpecialistApplication

Поле	Тип	Nullable
id	int	ні
user_id	string / int	ні
admin_id	string / int	так
status	string	ні
comment	string	так
created_at	string (datetime)	ні
updated_at	string (datetime)	ні

Таблиця 2.17

Таблиця Schedule

Поле	Тип	Nullable
id	int	ні
doctor_id	int	ні
datetime	string (datetime)	ні
duration	int	ні
details	string	так
client_id	int	так
status	string	так
meet_url	string	так
recommendations	string	так

Таблиця 2.18

Таблиця Review

Поле	Тип	Nullable
id	int	ні
doctor_id	int	ні
client_id	int	ні
rating	int	ні
comment	string	так
date	string (date)	ні

Таблиця 2.19

Таблиця Article

Поле	Тип	Nullable
id	int	ні
doctor_id	int	ні
created_at	string (datetime)	ні
updated_at	string (datetime)	ні
content	string	ні
status	string	ні

Центральною сутністю є таблиця User, яка містить основну інформацію про всіх користувачів системи. Вона включає унікальний ідентифікатор, ім'я, email, дату народження, роль, тип авторизації, токени доступу, а також допоміжні поля, як-от місто, адреса, стать та контактний номер. Особливістю є велика кількість полів у форматі JSON-масивів – мови, типи терапії, теми, підходи, формати, спеціалізації, досвід, соціальні мережі тощо. Це дозволяє зберігати складні структури даних і гнучко описувати професійний профіль. Додатково вказуються поля для освітніх документів, сертифікатів і фото. Таблиця також зберігає дату створення профілю та ключі авторизації. Вона реалізує інтерфейс автентифікації IdentityInterface і успадковує поведінку від ActiveRecord.

На основі таблиці User реалізовано таблицю SpecialistApplication, яка є розширенням і описує процес подачі заявки на статус спеціаліста. Вона містить зовнішній ключ до користувача, а також додаткові поля – статус заявки, коментар адміністратора, дату створення та оновлення, а також ідентифікатор адміністратора, який її обробив. Таким чином, структура дозволяє відслідковувати процес модерації спеціалістів у системі.

Таблиця Schedule відповідає за управління графіком консультацій. Вона містить дані про дату та час сеансу, тривалість, лікаря й клієнта, а також опціональні поля – статус зустрічі, посилання на відеозустріч, додаткові деталі та рекомендації. Завдяки цьому система може організовувати сесії між спеціалістами та клієнтами в задані часові слоти.

Модель Review дозволяє користувачам залишати відгуки на спеціалістів. Вона включає ідентифікатори лікаря й клієнта, оцінку, дату та текст коментаря. Ця таблиця дає змогу будувати прозору систему оцінювання послуг на платформі та формувати довіру між користувачами.

Окремо реалізовано таблицю Article, яка зберігає авторські публікації лікарів. Кожна стаття містить вміст, статус («reviewing», «approved», «rejected»), а також дату створення й оновлення. Вказується зв'язок з користувачем, який є її автором. Цей функціонал розширює платформу як джерело знань і самопрезентації фахівців.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

Щоб наочно представити логічну структуру даних, взаємозв'язки між ключовими таблицями, а також забезпечити основу для реалізації функціоналу відповідно до бізнес-логіки платформи, було побудовано діаграму БД (Рис. 2.3).

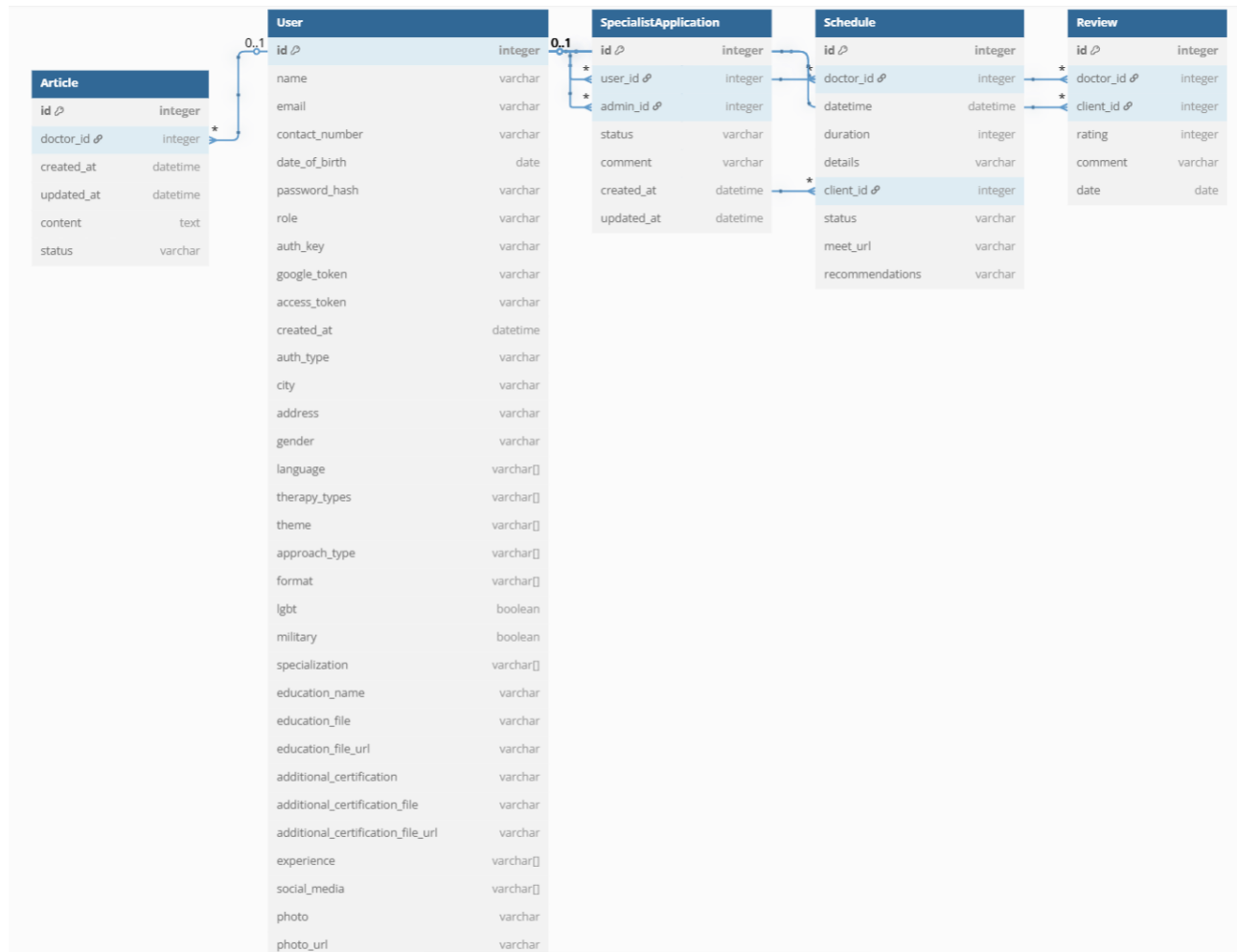


Рис. 2.3. Діаграма БД

На ній ми можемо побачити побудовані зв'язки між таблицями задля коректної роботи додатку.

Таблиця SpecialistApplication має зовнішній ключ user_id, що вказує на користувача, який подав заявку на приєднання як спеціаліст. Поле admin_id також посиляється на користувача з роллю адміністратора, який опрацьовує заявку. Таким чином реалізовано двосторонній зв'язок між спеціалістом та адміністратором у контексті обробки заявки.

Таблиця Schedule містить поле doctor_id, яке зв'язане з користувачем, що є лікарем, та client_id, що відповідає користувачу-пацієнту, який записався на

консультацію. Таким чином забезпечено реалізацію взаємодії між спеціалістом та клієнтом через розклад.

Таблиця Review також має два зовнішні ключі: doctor_id і client_id, що дозволяє зберігати відгуки клієнтів на консультації, проведені конкретними лікарями.

Таблиця Article пов'язана з таблицею User через поле doctor_id і зберігає публікації спеціалістів на платформі. Це забезпечує можливість ведення блогу та створення корисного контенту.

Завдяки таким зв'язкам система підтримує повноцінну багаторольову взаємодію між користувачами, організовує структуру даних відповідно до принципів нормалізації, та забезпечує цілісність інформації в межах платформи.

Загалом, структура бази даних забезпечує чітке розділення відповідальностей між користувачами й адміністрацією, підтримує повний цикл роботи з терапевтами – від заявки й публікацій до сесій, відгуків і комунікації. Вона також дозволяє масштабувати платформу завдяки підтримці складних типів даних, таких як масиви мов, підходів чи тем, зберігаючи при цьому гнучкість та простоту моделювання.

2.3. Проектування та реалізація алгоритмів роботи платформи

Алгоритм роботи додатку ілюструє послідовність дій і взаємодій між різними компонентами системи для досягнення конкретних результатів. Це може включати процеси автентифікації користувачів, обробки запитів, взаємодії з базою даних.

У рамках розробки системи «MindPlace» було побудовано діаграми активності, які відображають основні процеси системи, а саме авторизацію, отримання ролі лікаря, підбір спеціаліста, проведення консультації між лікарем та пацієнтом та окремо система блогу про ментальне здоров'я, що розміщений на цій же платформі. Діаграма активності системи «MindPlace» відображає послідовність дій у ключових сценаріях.

Розглянемо діаграму нижче (Рис. 2.4):

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

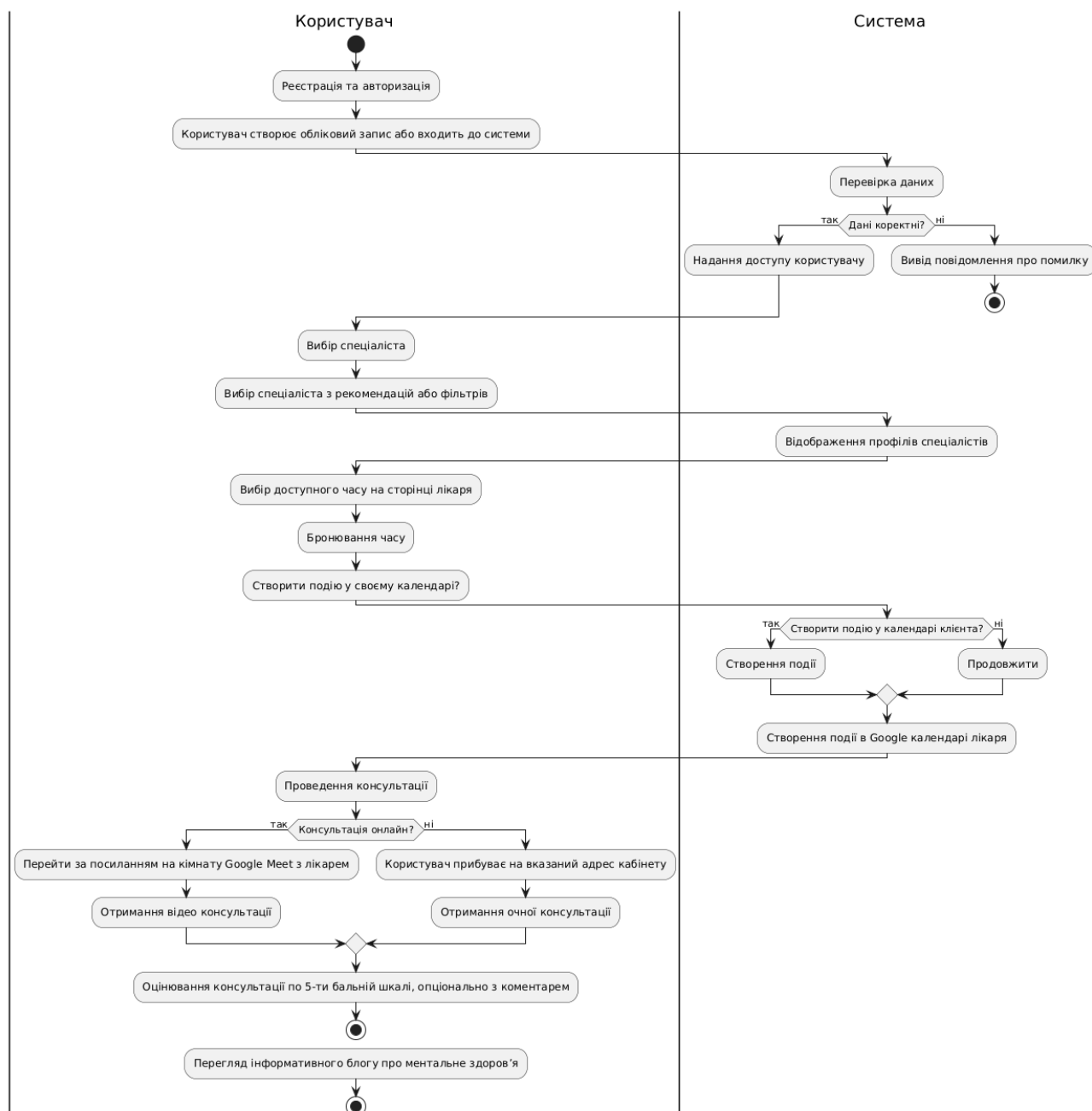


Рис. 2.4. Діаграма активності системи «MindPlace» для звичайного користувача

Користувач починає з реєстрації чи авторизації в системі, де вказує свою роль або входить за вже наданими даними. Після цього система перевіряє введену інформацію та надає доступ до сервісу. Далі він може переглядати активні профілі спеціалістів і вибирати того, хто найбільше відповідає його потребам, орієнтуючись на рекомендації або за допомогою доступних фільтрів. Після вибору спеціаліста користувач обирає зручний час для консультації, використовуючи доступну інформацію на сторінці лікаря і бронює його. Система, у свою чергу, створює подію в календарі лікаря нового запису. За годину до назначеного часу,

		Піщук С.О.		
		Чижмотря О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

лікар підтверджує створення посилання на кімнату Google Meet, що стає доступним користувачу та лікарю . Після завершення консультації пацієнт має змогу оцінити її якість за 5-бальною шкалою та залишити відгук. Крім того, пацієнт може переглядати блог про ментальне здоров'я для отримання корисної інформації.

Наступною наведено діаграму активності для спеціаліста (Рис. 2.5.):

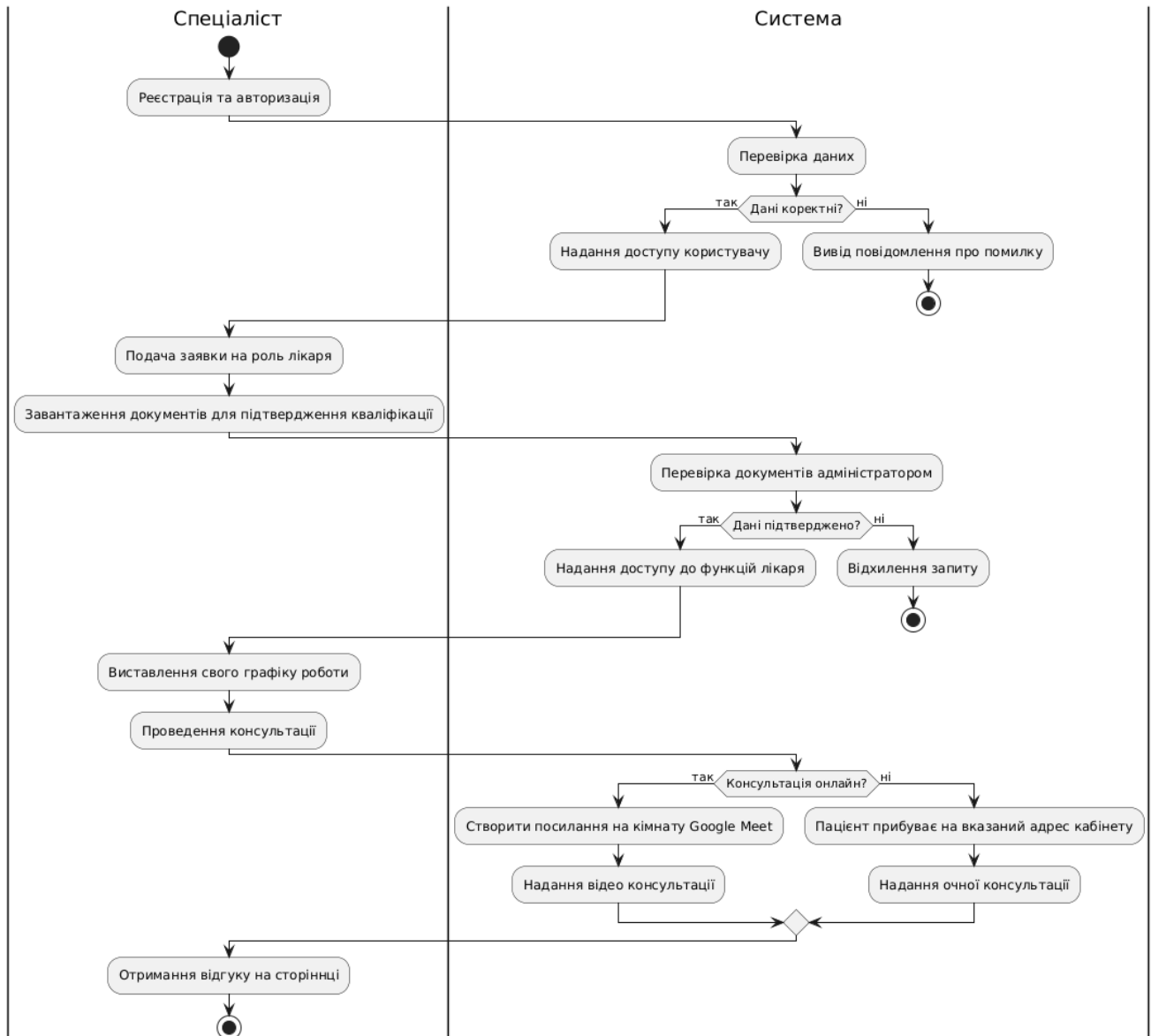


Рис. 2.5. Діаграма активності системи для спеціаліста

Спеціаліст починає з реєстрації чи авторизації в системі. Після цього цей користувач має можливість подати заявку на роль лікаря. Для цього він заповнює форму, завантажує документи, які підтверджують його кваліфікацію, і надсилає їх на перевірку адміністратору. Якщо документи відповідають вимогам, спеціаліст отримує доступ до функцій лікаря в системі; в іншому випадку його запит

відхиляється. Далі, лікар має можливість публікувати свій графік на сторінці, вибираючи доступний час для прийому пацієнтів. Коли пацієнти бронюють час, лікар проводить консультацію через відео-зв'язок або очно у кабінеті за адресом.

Крім того, для спеціалістів є можливість створювати та додавати статті до блогу про ментальне здоров'я. Лікар пише статтю та передає її на перевірку адміністраторам. Якщо стаття проходить перевірку, адміністратор публікує її на блозі; якщо ж ні, стаття повертається автору на редагування. У наступній діаграмі зображено ці події (Рис. 2.6.):

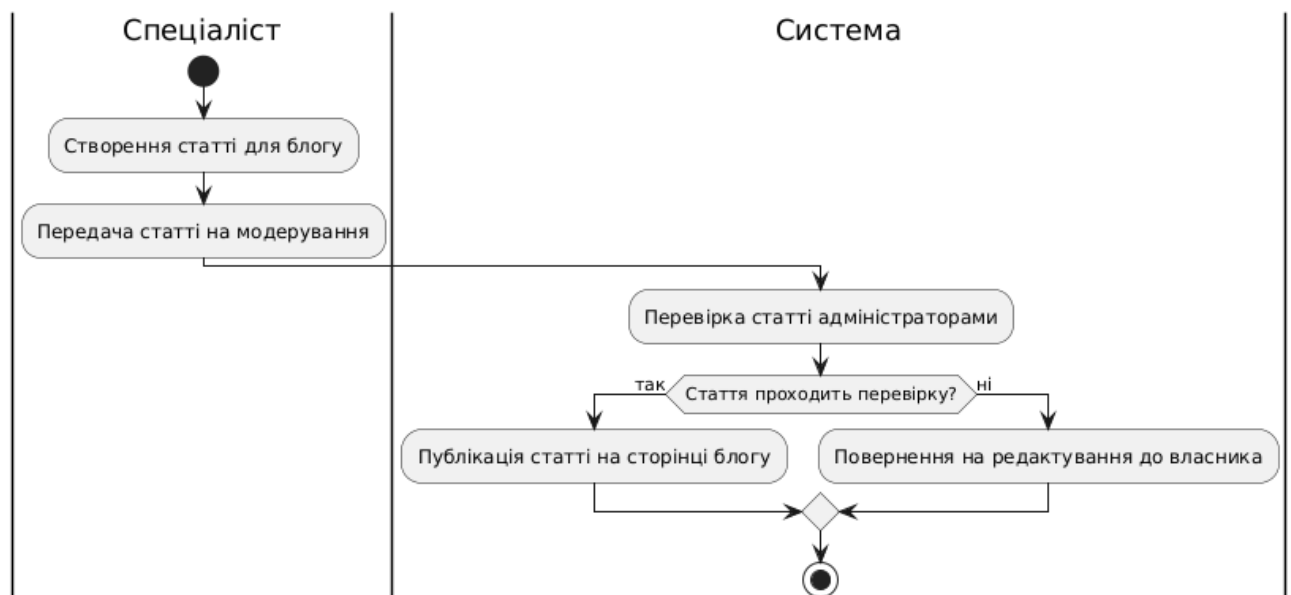


Рис. 2.6. Діаграма активності системи для спеціаліста щодо ведення блогу

Дані сценарії охоплюють всі основні функціональні процеси, що включають авторизацію, вибір спеціаліста, виконання запису на консультації, отримання зворотного зв'язку, процес присвоєння ролей користувачам. Окремо виділено сценарії роботи інформативного блогу та процес проведення консультацій в онлайн режимі. Це дозволяє чітко розуміти логіку послідовності дій у системі, відображати стани користувачів та забезпечувати їх ефективну взаємодію із функціоналом платформи.

2.4. Реалізація функціоналу додатку

Уся серверна логіка додатку ментального здоров'я MindPlace побудована з використанням фреймворку Yii2, що забезпечує ефективну обробку запитів та

розширюваність. У Yii2 додатки за замовчуванням реалізовані за патерном MVC – це основна архітектура фреймворку. У рамках цієї архітектури модель User реалізує як механізм доступу до бази даних через патерн ActiveRecord, так і логіку автентифікації користувача, завдяки реалізації інтерфейсу IdentityInterface. Це дозволяє моделі одночасно виконувати роль репрезентації даних і бути учасником механізму авторизації, що вбудовано у Yii2 через компонент Yii::\$app->user.

Взаємодія між контролерами, моделями і шаблонами побудована згідно з MVC-підходом: контролери приймають HTTP-запити, викликають відповідні методи моделі, а потім передають результати до виглядів (views), що відповідають за візуальне представлення. При цьому модель може містити бізнес-логіку, зокрема, валідацію введених даних, обробку аутентифікації, створення та оновлення облікових записів.

У майбутньому, з метою підтримки чистішої архітектури та кращого розділення обов'язків, передбачено поступовий перехід до використання додаткових рівнів абстракції, зокрема сервісного шару, що дозволить винести частину логіки з моделей у окремі сервіси. Це сприятиме масштабованості проєкту та спрощенню тестування окремих компонентів.

Центральними елементами платформи є модулі перевірки кваліфікації лікарів та процесу проведення консультації.

Отже весь процес подання заявки проходить на одній сторінці сайту, де ініціалізується модель TherapistJoinForm. Поступово при заповненні підвантажуються відповідні частини форми (особиста інформація, підходи до терапії, освіта, завантаження документів). Кожен крок зберігається в сесію, щоб зібрати всі дані до фінального сабміту, а самі вони провантажуються через AJAX [20], що повертає renderPartial() потрібного _view з вкладеною моделлю.

Для збереження файлів обрано варіант використання хмарного сховища Amazon S3 [21], що у системі подачі заявки терапевта має стратегічне значення, особливо в контексті обробки та зберігання особистих і чутливих файлів користувачів, таких як фото, дипломи та сертифікати. У наведеному нижче коді

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

реалізовано окремий компонент S3Storage, який інкапсулює логіку підключення до AWS S3, перевірку доступності клієнта, завантаження, видалення та тестування з'єднання з бакетом. Ініціалізація клієнта представлена методом getClient():

```
public function getClient()
{
    if (!$this->_client) {
        if (!$this->key || !$this->secret) {
            Yii::warning('AWS credentials not set properly', 'therapist-join');
            return null;
        }

        $this->_client = new S3Client([
            'version' => 'latest',
            'region' => $this->region,
            'credentials' => [
                'key' => $this->key,
                'secret' => $this->secret,
            ],
        ]);

        return $this->_client;
    }
}
```

Основна перевага використання S3 полягає у високій надійності, масштабованості та доступності даних. Хмара зберігає файли поза межами локального сервера, що суттєво зменшує ризик втрати даних у разі відмови хостингу або контейнера. Крім того, це дозволяє швидко масштабувати систему: незалежно від кількості заявок або розміру файлів, S3 автоматично обробить навантаження.

Компонент S3Storage ініціалізується з ключами доступу AWS (\$key, \$secret), назвою бакету (mindplacediploma) та регіоном розгортання (eu-north-1). Метод getClient() забезпечує ліниву ініціалізацію клієнта S3 лише при потребі, водночас перевіряючи валідність облікових даних. Це дозволяє уникнути зайвих помилок при кожному запиті та централізовано логувати проблеми підключення.

Завантаження файлів відбувається через метод upload(), який приймає або шлях до файлу, або об'єкт UploadedFile. Система відкриває файл як потік і передає його в putObject(), встановлюючи правильний ContentType, що важливо для безпечної й коректної передачі документів. Після завантаження повертається URL об'єкта, що дозволяє одразу використовувати посилання для подальшого перегляду

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

або перевірки з боку адміністратора. Лістинг реалізованого коду цього методу нижче:

```
public function upload(UploadedFile|string $file, string $path)
{
    if (!$this->client) {
        return false;
    }

    if (is_string($file)) {
        $stream = fopen($file, 'rb');
    } else {
        $stream = fopen($file->tempName, 'rb');
    }

    $result = $this->client->putObject([
        'Bucket' => $this->bucket,
        'Key'     => $path,
        'Body'    => $stream,
        'ContentType' => mime_content_type($file),
    ]);

    fclose($stream);

    return [
        'name' => basename($file),
        'path' => $path,
        'url' => $result['ObjectURL'] ?? null,
    ];
}
```

У формі TherapistDocumentsForm є метод uploadToS3(), який завантажує файли користувача до S3 в персоналізовані директорії (users/{user_id}/documents або /photo). Це дозволяє логічно структурувати зберігання й легко керувати файлами – наприклад, видаляти їх, якщо заявка відхилена, або надавати тимчасовий доступ. Крім того, файли спочатку зберігаються у тимчасову директорію @runtime/uploads, а після передачі на S3 – одразу видаляються з сервера (unlink(...)). Це зменшує навантаження на сервер і запобігає накопиченню тимчасових файлів, які можуть містити чутливу інформацію. Нижче наведено до прикладу лістинг частини методу uploadToS3, який демонструє створення іменованих директорій, вивантаження файлів на хмару та видалення тимчасових файлів стосовно документу про вищу освіту:

```
$userId = Yii::$app->user->id;
$directory = "users/{$userId}/documents";
$photo_directory = "users/{$userId}/photo";

$result = [];
```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

try {
    if ($this->education_file_path && file_exists($this->education_file_path)) {
        $result['education_file'] = $s3->upload(
            $this->education_file_path,
            $directory . '/education/' . time() . '_' . basename($this->education_file_path)
        );
        unlink($this->education_file_path);
    }
}

```

Продовжуючи загальну специфіку інтеграції з Amazon S3, у проєкті реалізовано також механізм завантаження файлів із хмарного сховища, що дозволяє адміністраторам або іншим уповноваженим особам отримувати доступ до збережених раніше документів користувачів. Це важливо для перевірки достовірності заявок, верифікації дипломів, сертифікатів або фотографій.

Завантаження організовано через метод `actionDownload($file)`, який приймає повну URL-адресу або частковий шлях до об'єкта в S3. Ключовим моментом у завантаженні є метод `extractKeyFromUrl(string $url)`, який витягує відносний шлях об'єкта в бакеті з повної URL-адреси. Це необхідно, оскільки S3 API вимагає для завантаження або видалення не повну URL-адресу файлу, а саме значення Key – тобто шлях до об'єкта в межах бакету. Для того щоб передати вміст файлу користувачеві, використовується `Yii::$app->response->sendStreamAsFile(...)`, який дозволяє відправити тіло файлу в HTTP-відповіді як окремий файл, що буде автоматично збережений або відкритий браузером. Реалізація методу продемонстрована у наступному лістингу:

```

public function actionDownload($file)
{
    $s3 = Yii::$app->get('s3Storage');
    if (!$s3 || !$s3->client) {
        Yii::error('S3 клієнт не ініціалізовано', 'therapist-join');
        Yii::$app->session->setFlash('error', 'Сховище недоступне.');
```

```

        return $this->redirect(Yii::$app->request->referrer);
    }

    try {
        $key = $this->extractKeyFromUrl($file);

        $result = $s3->getClient()->getObject([
            'Bucket' => $s3->bucket,
            'Key' => $key,
        ]);

        $stream = $result['Body']->detach();
        $filename = basename($key);
    }
}

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return Yii::$app->response->sendStreamAsFile($stream, $filename);
    } catch (\Exception $e) {
        Yii::error("Download error: " . $e->getMessage(), 'admin.download');
        Yii::$app->session->setFlash('error', 'Не вдалося завантажити файл.');
```

Додатково реалізовано окремі методи на завантаження конкретного виду файлу освіти/фото, що розмежовують запити користувача та делегують завантаження до основного методу завантаження, показаного раніше.

Загалом, використання S3 дозволяє розділити зберігання контенту від основного застосунку, забезпечує стабільність, централізованість та захищеність обробки документів. У межах вебплатформи ментального здоров'я це також підвищує довіру користувача до сервісу – завдяки гарантованому зберіганню важливих документів.

Інтеграція системи з Google Calendar і Google Meet у цьому проєкті реалізована як додатковий сервіс, який автоматично створює події для запланованих консультацій на основі розкладу, надає зручні посилання для доступу до відеосесій і синхронізує це з Google-акаунтами фахівців. Основна ідея – централізувати управління сесіями та мінімізувати ручні дії як для терапевта, так і для пацієнта, водночас забезпечуючи високу надійність за рахунок використання API Google.

Серцем інтеграції є клас GoogleClient, який налаштовує підключення до Google API [22]. Він використовує файл облікових даних google-credentials.json та зберігає access token у google-token.json у директорії runtime. Це дозволяє працювати з календарем Google у від імені застосунку або користувача з мінімальним втручанням після первинної авторизації.

Після успішного підключення, події можуть додаватися до календаря за допомогою виклику команди. Наприклад, метод addToCalendar(...) створює стандартну подію на основі дати, часу і тривалості, додаючи її до календаря лікаря. Він також формує опис та заголовок події динамічно на основі імені лікаря.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

Система перевіряє, чи токен дійсний, і лише тоді виконує запит — таким чином уникнуто зайвих помилок.

Інша важлива частина — це створення зустрічей Google Meet. Метод `actionGmeet()` ініціює створення події з активованим блоком `conferenceData`, що автоматично додає посилання на Google Meet. Ця функціональність активується через параметр `conferenceDataVersion` при вставці події. Після створення події система отримує посилання, зберігає його в базі через метод `Schedule::updateMeetLink(...)` і повертає користувача на сторінку перегляду сесії з відповідним повідомленням.

Загалом, цей підхід дозволяє уникнути дублювання даних між системами, створює цілісну цифрову взаємодію без необхідності вручну копіювати дату або посилання на сесію, і забезпечує прозорий облік усіх подій. Вся інтеграція побудована на принципі сервісної взаємодії: Google виступає як зовнішній провайдер подій, а внутрішня система MindPlace — як координатор сесій і користувач API. Це дозволяє легко масштабувати функціональність (наприклад, додати скасування або оновлення подій), адаптувати логіку під персональні календарі кожного фахівця та підтримувати цілісну цифрову екосистему ментальної підтримки.

Висновки до другого розділу

У другому розділі докладно висвітлено архітектурне й функціональне проектування додатку для підтримки ментального здоров'я MindPlace, створеного з використанням патерну MVC на основі PHP та фреймворку Yii2. Визначено основні компоненти системи та розкрито принципи їх взаємодії. Розглянуто можливі сценарії використання платформи з урахуванням ролей користувача, адміністратора й спеціаліста, а також типові моделі їх взаємодії із системою.

Проектований функціонал охоплює надання психологічної підтримки, індивідуальний підбір спеціаліста, створення й перегляд медичних статей, що свідчить про фокус платформи на зручність, доступність та ефективність допомоги в питаннях ментального здоров'я.

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

Класова діаграма та структура бази даних ілюструють об'єктно-орієнтований підхід до побудови системи, що забезпечує її масштабованість і гнучкість при роботі з основними сутностями. Такий підхід дозволяє чітко організувати логіку взаємодії користувачів, модерацію та управління контентом.

Діаграми активності допомогли деталізувати ключові алгоритми – зокрема, процеси реєстрації, авторизації, процесу отримання та надання консультацій, маніпуляції над контентом. Це забезпечує прозоре розуміння функціонування платформи як для користувачів, так і для розробників.

У результаті сформовано логічно структуровану систему з чітким розподілом відповідальностей між її компонентами. Така архітектура сприяє підтримці чистоти коду, легкості в обслуговуванні та можливості безперешкодного розширення функціоналу в майбутньому.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ДОДАТКОМ МЕНТАЛЬНОГО ЗДОРОВ'Я «MINDPLACE»

3.1. Порядок встановлення та налаштування параметрів системи

Інформаційна система MindPlace розроблена як веб-застосунок для підтримки ментального здоров'я користувачів і взаємодії з фахівцями – психологами та психотерапевтами. Система реалізована у вигляді багаторівневого веб-додатку з клієнтською частиною (frontend), серверною частиною (backend), базою даних та інтеграцією з зовнішніми сервісами, зокрема Google Calendar та Google Meet. Проєкт побудований на фреймворку Yii2 (PHP) зі стороною адміністратора, кабінетом спеціаліста, користувацьким інтерфейсом та API-ендпоінтами.

Для локального розгортання або встановлення на серверному середовищі необхідно забезпечити підтримку таких технологій:

- Docker та Docker Compose (версії 20+);
- Git для клонування репозиторію[23];
- Доступ до змінних середовища та .env-файлів;
- AWS CLI (для завантаження та обробки документів через Amazon S3).

Середовище розгортання контейнеризоване за допомогою Docker Compose, що дозволяє підняти сервіси з мінімальними налаштуваннями. Основні сервіси, передбачені у docker-compose.yml, включають: nginx – як веб-сервер [24], php – інтерпретатор із підтримкою Yii2, mysql — для зберігання даних, AWS S3 – для роботи з файлами (документи, дипломи тощо).

Після клонування репозиторію з Git необхідно відредагувати файл .env з урахуванням параметрів бази даних, шляхів до S3 та ключів OAuth Google API.

Далі виконується запуск сервісів через команду docker-compose up -d --build.

Після запуску контейнерів необхідно ініціалізувати базу даних шляхом застосування міграцій: docker exec -it php yii migrate.

Для взаємодії з Google Meet та Google Calendar реалізовано сервіс інтеграції, який використовує офіційні бібліотеки Google API на PHP. Авторизація

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

виконується за допомогою OAuth2 [30], токени зберігаються в базі даних. Після підключення календаря спеціаліст може створювати події у Google Calendar, що автоматично генерують посилання на відеозустріч через Google Meet.

Окрему увагу було приділено реалізації блогу на платформі. Для зручного редагування вмісту статей інтегровано редактор TinyMCE [25], який дозволяє спеціалістам створювати структуровані, форматовані тексти з підтримкою стилів, заголовків, посилань, списків, таблиць та зображень. Збереження таких статей відбувається у форматі HTML, що дозволяє відображати їх із збереженням форматування в публічному блозі. Для безпечного збереження вмісту реалізовано фільтрацію HTML-тегів та обмеження допустимих елементів. Це дозволяє поєднати гнучке візуальне редагування з безпечним зберіганням та відображенням контенту.

Інтерфейс користувача доступний через браузер за адресою: <http://localhost:8080/> де користувачі можуть авторизуватись відповідно до своєї ролі (звичайний користувач, спеціаліст, модератор або адміністратор). Спеціалісти можуть проходити процес приєднання, додавати свої сертифікати, проходити модерацію, керувати календарем записів і бачити заплановані зустрічі. Адміністратор має доступ до панелі керування заявками спеціалістів, аналітики, керування користувачами та логами.

Таким чином, MindPlace є повноцінною сучасною багаторівневою системою, яку можна запускати як у локальному середовищі для розробки, так і на віддалених серверних потужностях, або у хмарній інфраструктурі за допомогою контейнеризації. Це дозволяє зберігати стабільність, масштабованість і гнучкість під час експлуатації.

3.2. Структура інтерфейсу додатку ментального здоров'я

MindPlace має адаптивний, інтуїтивно зрозумілий інтерфейс, побудованим за сучасними принципами UX/UI з використанням компонування Bootstrap та кастомного CSS. Платформа підтримує мультирольову автентифікацію: звичайні

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

користувачі (пацієнти), спеціалісти (терапевти) та адміністратори. Вхід можливий за допомогою email/паролю або через OAuth (Google).

Користування сайтом розпочинається зі сторінки входу (Рис. 3.1), де користувач може авторизуватись традиційно або скористатися входом через Google. У разі вибору другого варіанту система перенаправляє на авторизацію Google (Рис. 3.2), після чого повертає користувача до системи.

Рис. 3.1. Сторінка входу в акаунт

Рис. 3.2. Перенаправлення при вході через Google

Нові користувачі можуть створити обліковий запис через сторінку реєстрації (Рис. 3.3).

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.3. Сторінка реєстрації нового акаунту

Після входу користувачі потрапляють на головну сторінку сайту (Рис. 3.4), яка містить короткий огляд функцій платформи та навігацію. Інтерфейс підтримує перемикання мови, зокрема є англійська локалізація (Рис. 3.5).

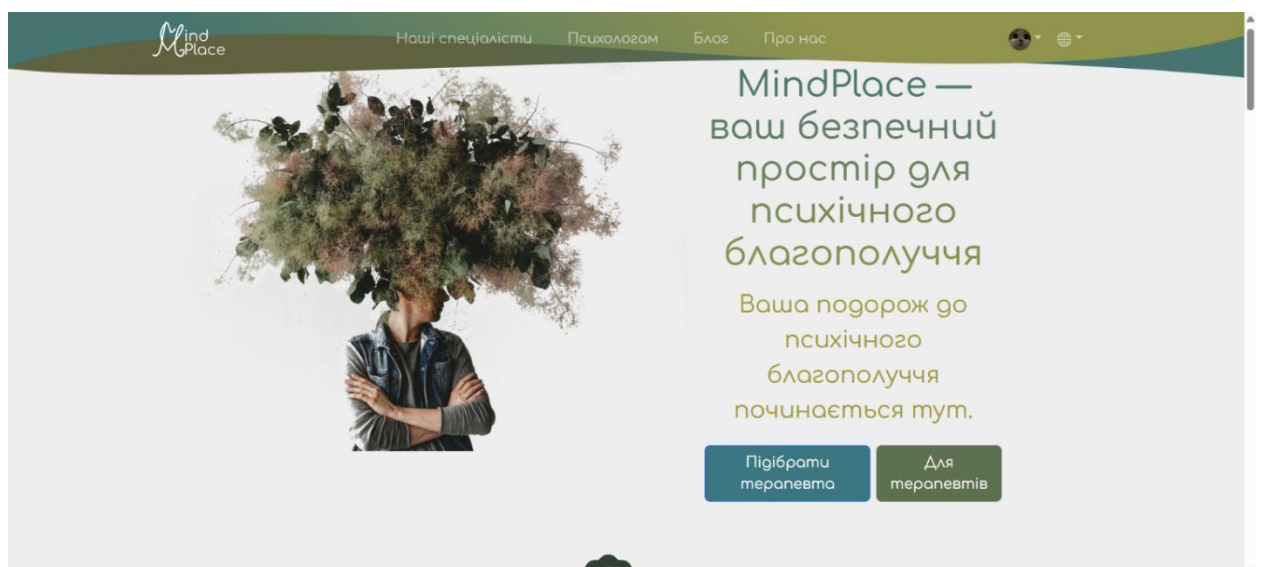


Рис. 3.4. Головна сторінка сайту

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

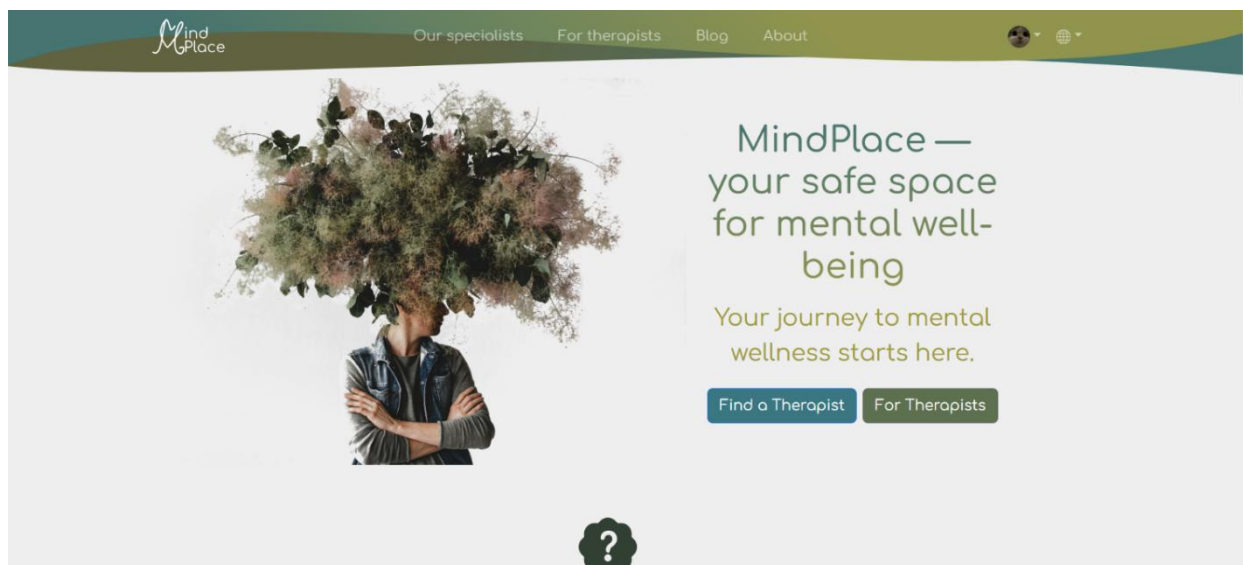


Рис. 3.5. Англійська локалізація сайту

Увійшовши до системи, звичайний користувач переходить у свій профіль, де відображається базова інформація про акаунт (Рис. 3.6).

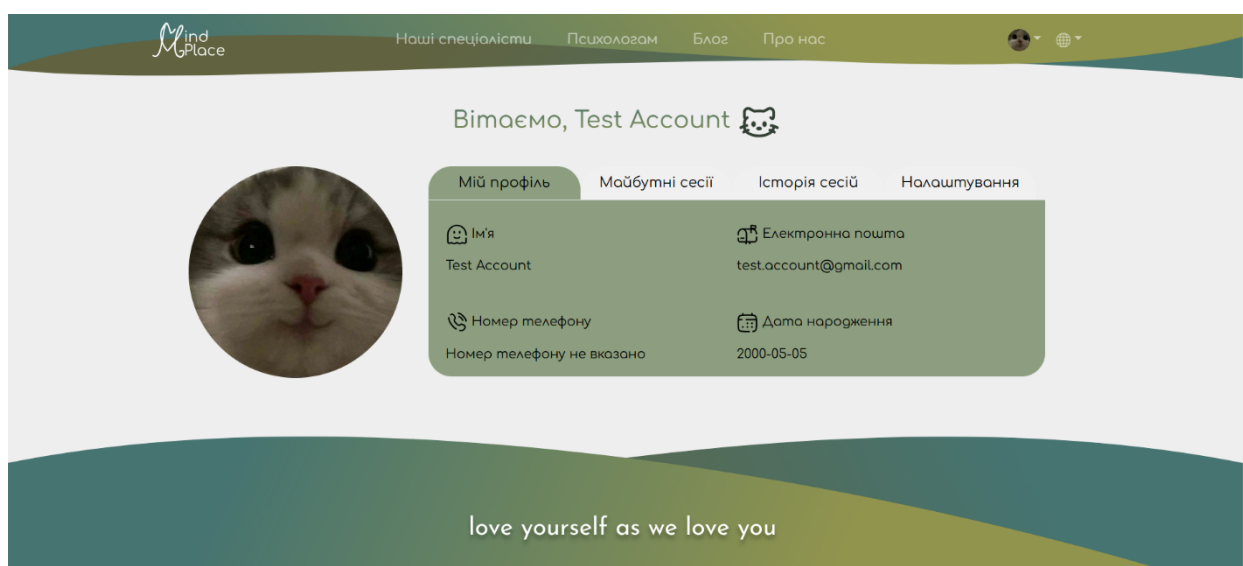


Рис. 3.6. Сторінка профілю звичайного користувача (дані акаунту)

В окремих вкладках доступні: список майбутніх консультацій з деталями та можливістю скасування (Рис. 3.7); історія отриманих консультацій та записами сесій (Рис. 3.8).

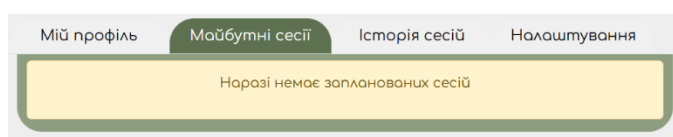


Рис. 3.7. Сторінка профілю звичайного користувача (записи на майбутні консультації)

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

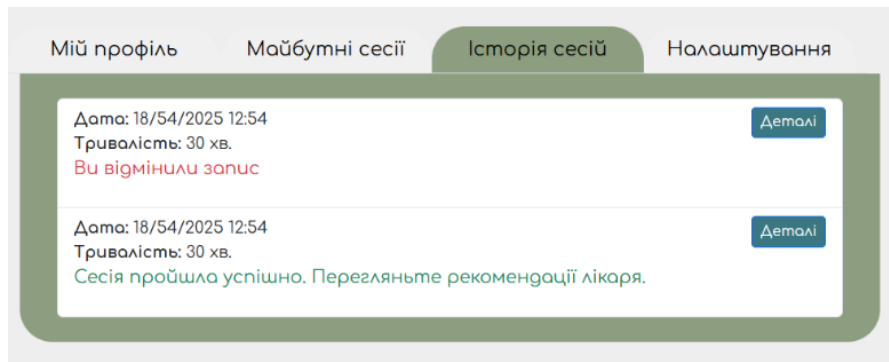


Рис. 3.8. Сторінка профілю звичайного користувача (історія отриманих консультацій)

Також користувачу доступно налаштування власного профілю, зокрема зміна персональних даних, пароля та інших параметрів (Рис. 3.9).

Рис. 3.9. Сторінка профілю звичайного користувача (налаштування, зміна даних)

Користувач може переглядати список доступних терапевтів на відведеній сторінці за допомогою фільтрації за спеціалізацією, підходами, мовою тощо (Рис. 3.10). Перейшовши до профілю терапевта, можна побачити загальну інформацію, спеціалізацію та досвід лікаря (Рис. 3.11).

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

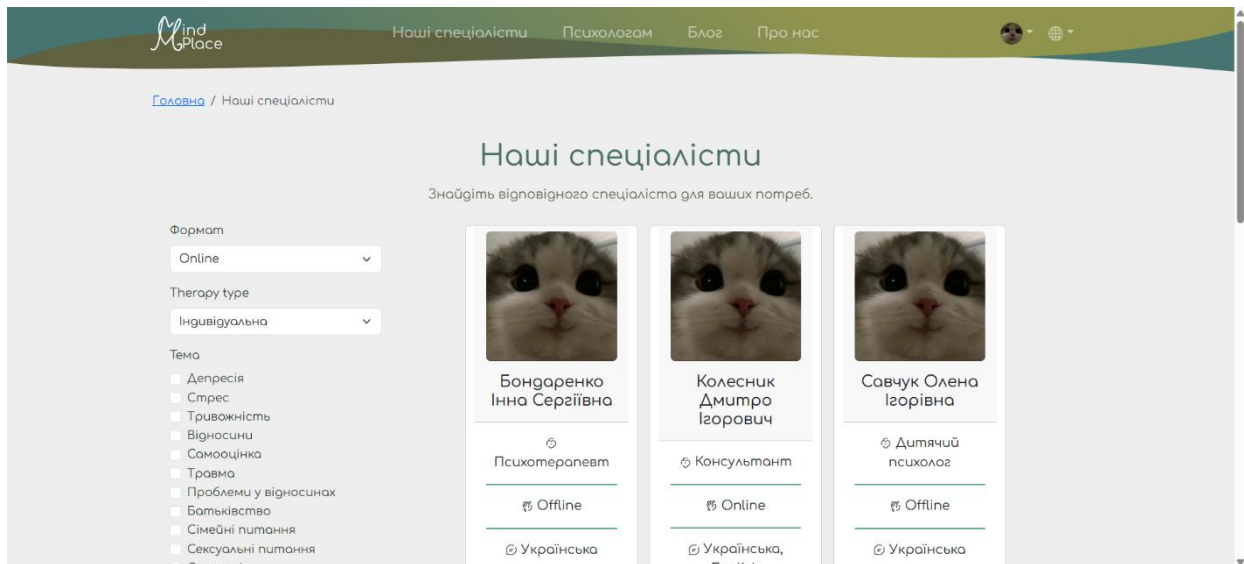


Рис. 3.10. Сторінка списку наявних спеціалістів з фільтром

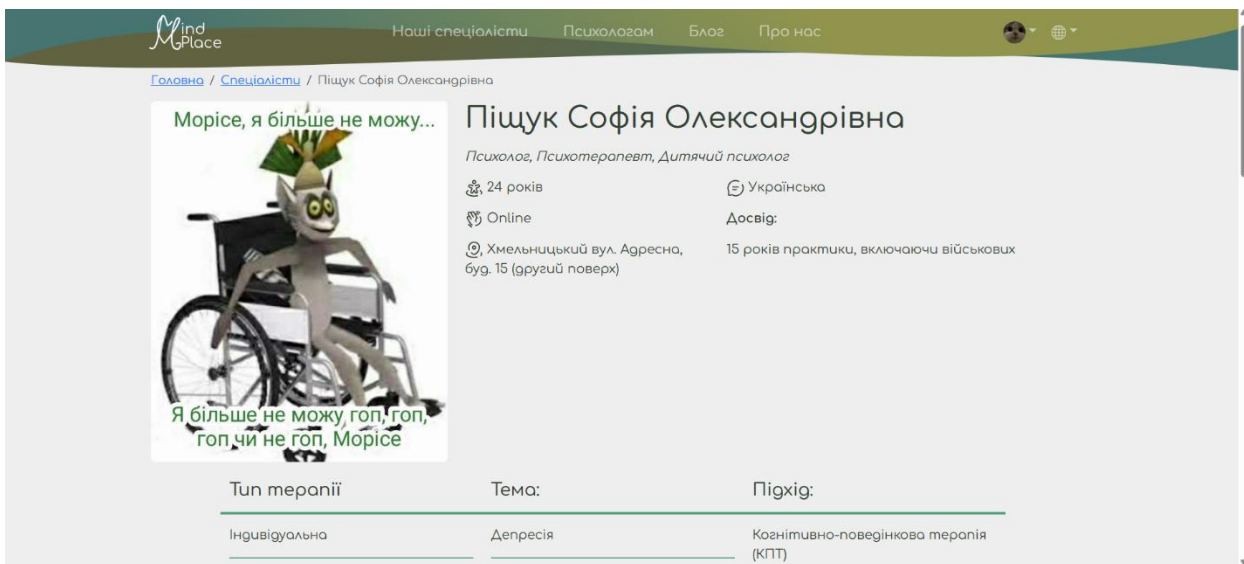


Рис. 3.11. Сторінка спеціаліста (загальна та професійна інформація)

Якщо цей спеціаліст підходить користувачу, він переходить до перегляду секції доступних слотів для бронювання сесій (Рис. 3.12). На вкладці відгуків користувачі можуть прочитати думки інших клієнтів або залишити власну, але тільки якщо вже має завершену консультацію з конкретно цим спеціалістом (Рис. 3.13).

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

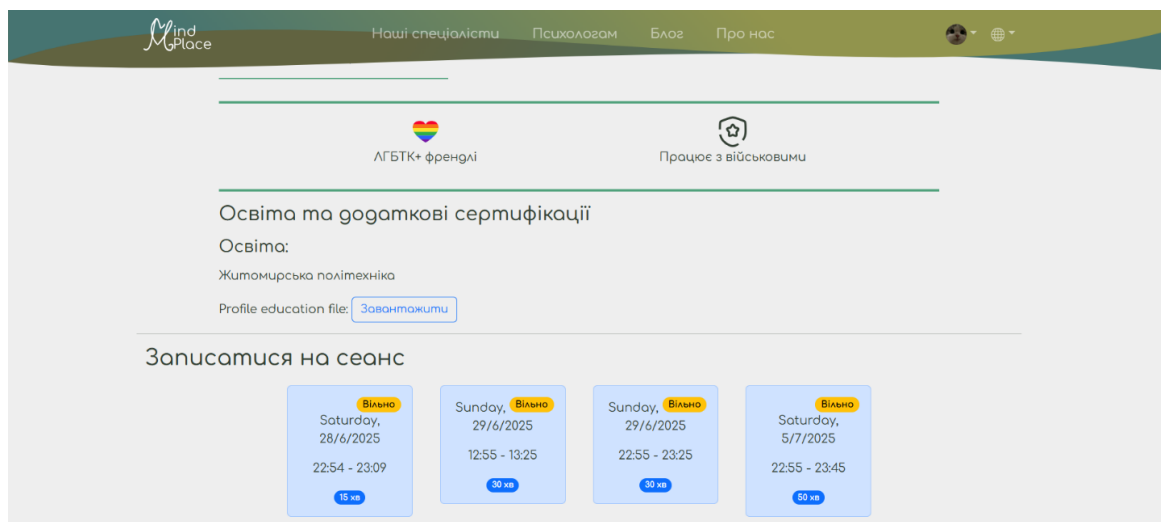


Рис. 3.12. Сторінка спеціаліста (слоти бронювання часу сесії)

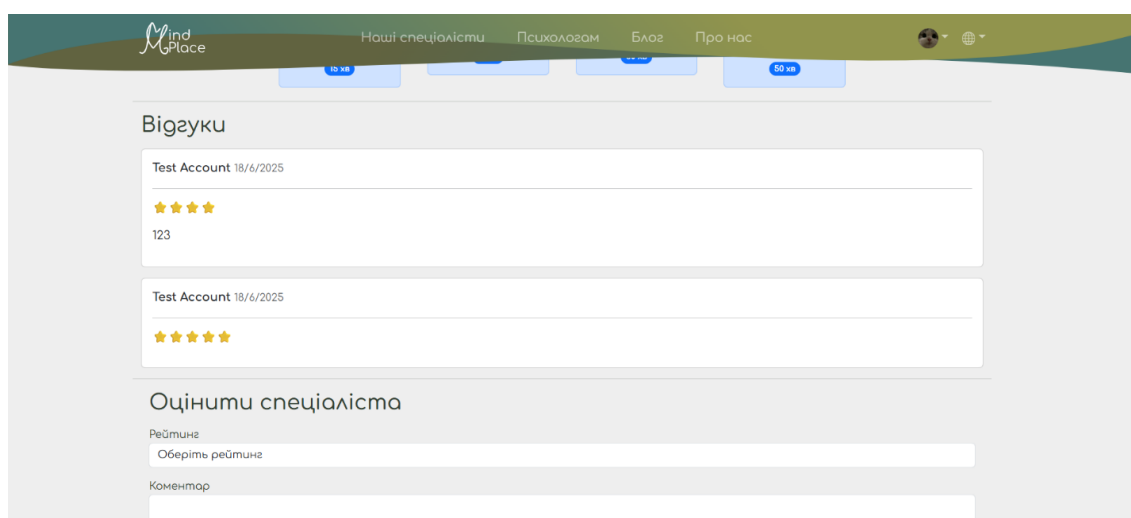


Рис. 3.13. Сторінка спеціаліста (відгуки та форма для рев'ю)

Процес запису складається з двох частин: перегляд загальної інформації про обрану сесію (Рис. 3.14) та заповнення форми з деталями (Рис. 3.15).

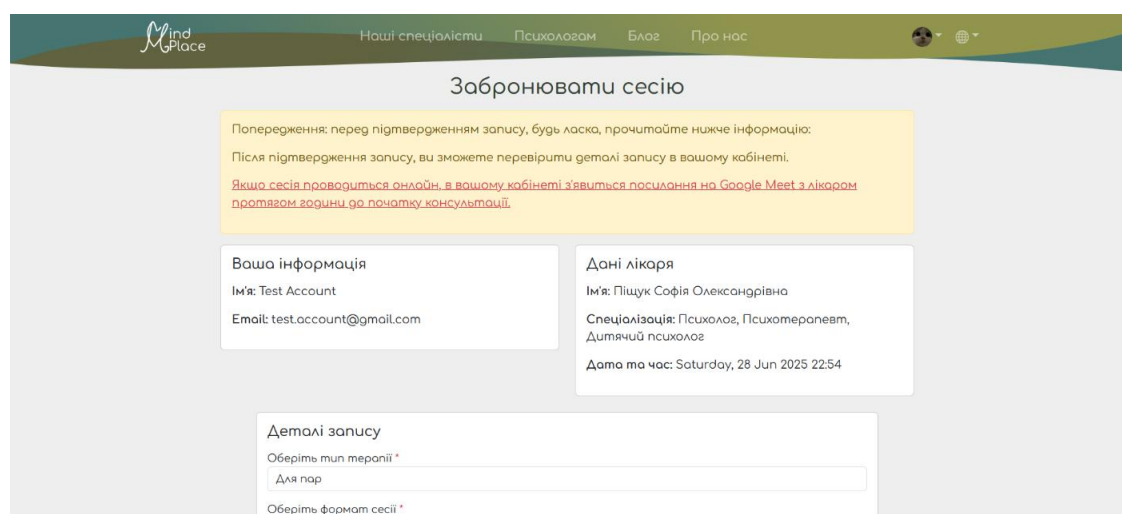


Рис. 3.14. Сторінка бронювання сесії (загальна інформація)

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.15. Сторінка бронювання сесії (форма деталей запису)

Після підтвердження, користувач може переглядати деталі консультації у двох форматах – з повним описом спеціаліста, часу, тощо (Рис. 3.16-3.17).

Рис. 3.16. Сторінка деталей консультації (1)

Рис. 3.17. Сторінка деталей консультації (2)

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

Крім того, у кабінеті лікаря реалізовано створення посилань на Google Meet кімнату для консультації. Нижче продемонстровано інтерфейс генерації посилання (Рис. 3.18), після чого воно стає доступне в кабінеті лікаря, та посилання доступне для пацієнта (Рис. 3.19). Створення посилання стає доступне за годину до початку запису.

Деталі консультації

№3 Скасувати

Спеціаліст: Психолог, Психотерапевт, Дитячий психолог — [Піщук Софія Олександрівна](#)

Клієнт: Test Account

Дата та час: Sunday, 22 Jun 2025, 21:00 – 21:15

Тривалість: 15 хв

Статус: Зареєстровано

Створити Google Meet

Рис. 3.18. Створення посилання на Google Meet (кабінет лікаря)

Деталі консультації

№3 Скасувати

Спеціаліст: Психолог, Психотерапевт, Дитячий психолог — [Піщук Софія Олександрівна](#)

Клієнт: Test Account

Дата та час: Sunday, 22 Jun 2025, 21:00 – 21:15

Тривалість: 15 хв

Статус: Заброньовано

Посилання на зустріч: <https://meet.google.com/myz-jtms-kth>

Рис. 3.19. Посилання створено (кабінет пацієнта)

Подія в календарі лікаря створюється обов'язково, але для цього йому потрібно увійти в свій кабінет через Google акаунт (Рис. 3.19). В календарі пацієнта вона створюється за його бажанням (поставити відмітку у формі «Додати подію в мій Google Calendar»), але також тільки якщо підключений його Google акаунт та надано доступ до маніпуляцій з його календарем.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

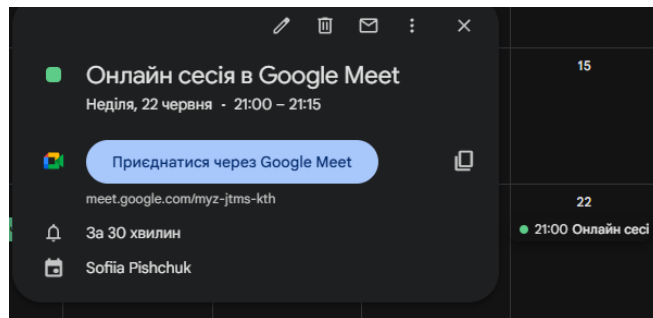


Рис. 3.20. Подія в Google Calendar лікаря

Спеціалісти, які бажають приєднатися до платформи, заповнюють покрокову форму заявки (Рис. 3.21). Вона включає кілька етапів. Першим етапом є заповнення особистої інформації (ПІБ, email, номер телефону, дата народження, стать, місто, опціонально посилання на соціальні мережі) (Рис. 3.22);

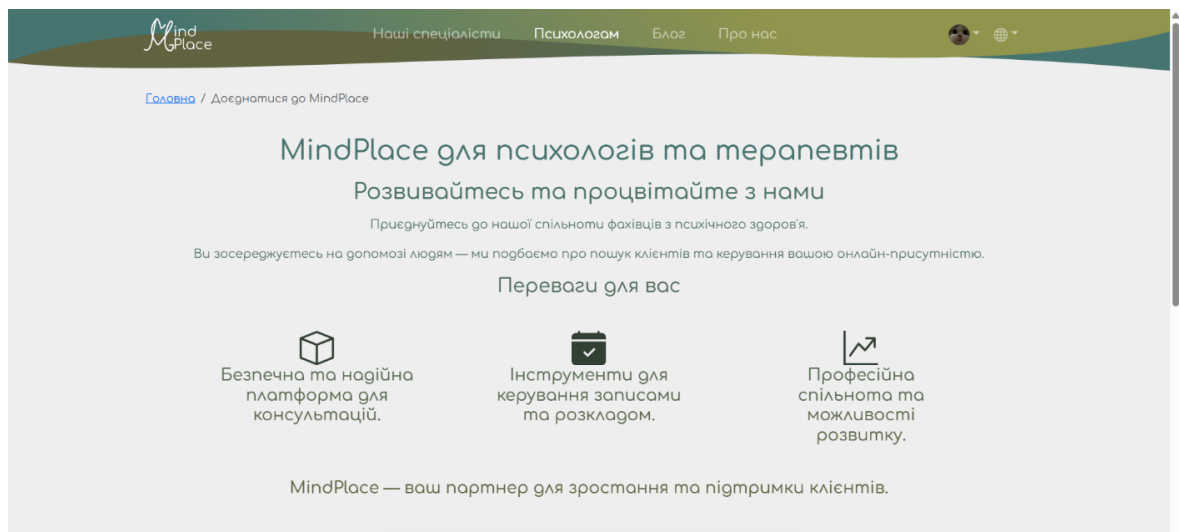


Рис. 3.21. Сторінка для терапевтів які хочуть доєднатися

Рис. 3.22. Крок «особиста інформація» форми подачі заявки

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

Другим етапом є форма «специфіка терапії» (методи, підходи, спосіб проведення, цільова аудиторія) (Рис. 3.23).

Специфіка терапії

Мова *

☐ Українська
☐ English

Тип *

☐ Індивідуальна
☐ Для пар
☐ Коучинг
☐ Діти (до 12 років)
☐ Підлітки (12-18 років)

Підхід до терапії *

☐ Когнітивно-поведінкова терапія (КПТ)
☐ Гештальт-терапія
☐ Психоаналіз
☐ Арт-терапія

Тема *

☐ Депресія
☐ Стрес
☐ Тривожність
☐ Відносини
☐ Самооцінка

Рис. 3.23. Крок «специфіка терапії» форми подачі заявки

У третьому етапі спеціаліст зазначає місця здобуття освіти, спеціалізацію та досвід роботи у цій сфері (Рис. 3.24).

Освіта та досвід

Назва навчального закладу *

e.g. University Name

Спеціалізація *

☐ Психолог
☐ Психотерапевт
☐ Дитячий психолог
☐ Консультант

Additional certification

e.g. Courses Name

Досвід *

← → **B** *I* U

-

-

e.g. Place, position: 5 years

Рис. 3.24. Крок «освіта та досвід» форми подачі заявки

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Четвертий і останній етап є дуже важливим, адже він включає надсилання документів підтверджуючих кваліфікацію та власне фото (Рис. 3.25).

Рис. 3.25. Крок «документи» форми подачі заявки

Після завершення процесу з'являється повідомлення про успішне надсилання заявки (Рис. 3.26).

Рис. 3.26. Повідомлення про успішне надсилання заявки

Поки заявка перебуває на розгляді, спеціаліст бачить свій профіль у режимі перегляду (Рис. 3.27), а функція створення графіка залишається недоступною (Рис. 3.28).

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				53
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 3.27. Сторінка свого профілю доки вона на розгляді

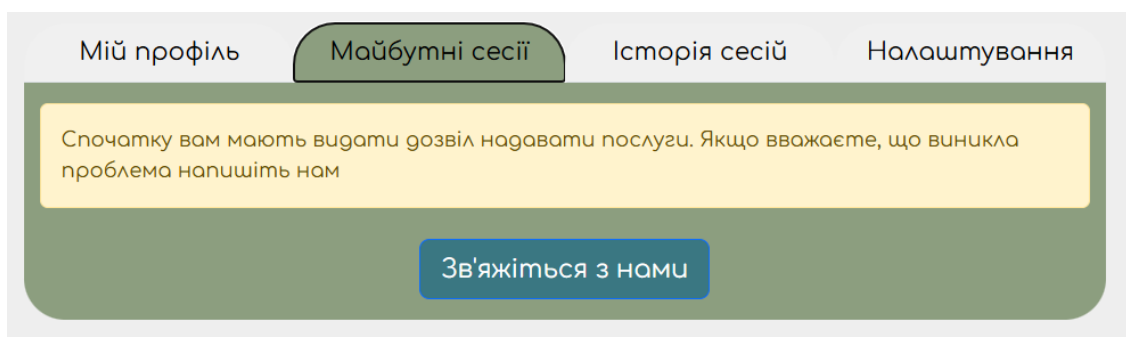


Рис. 3.28. Неможливість створювати графік доки заявка не підтверджена

Адміністратор має доступ до окремої панелі керування (Рис. 3.29), де здійснюється перегляд, обробка та модерація заявок спеціалістів.

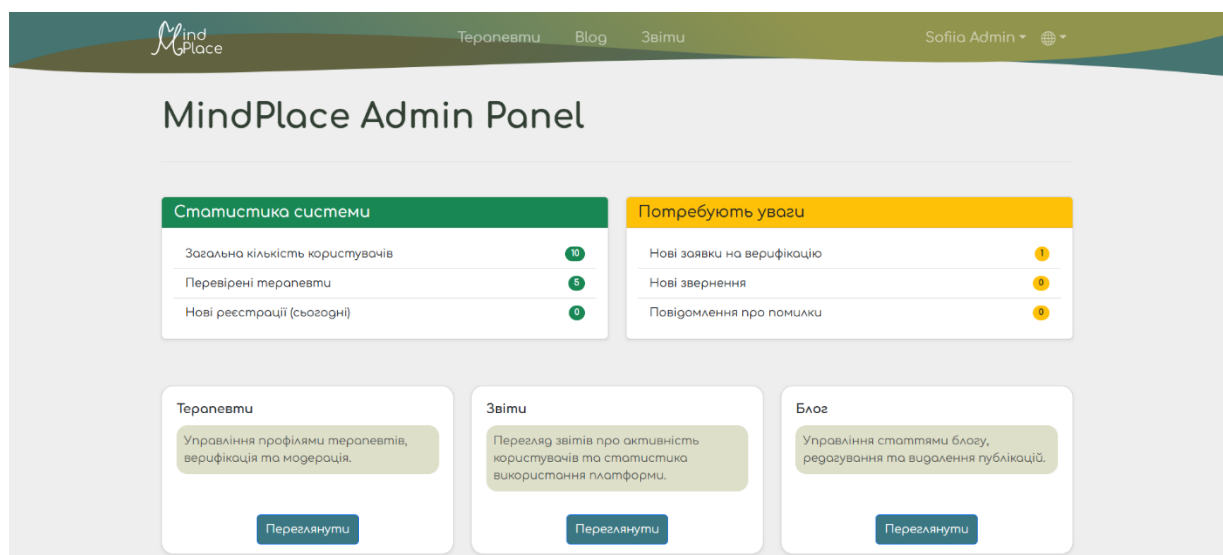
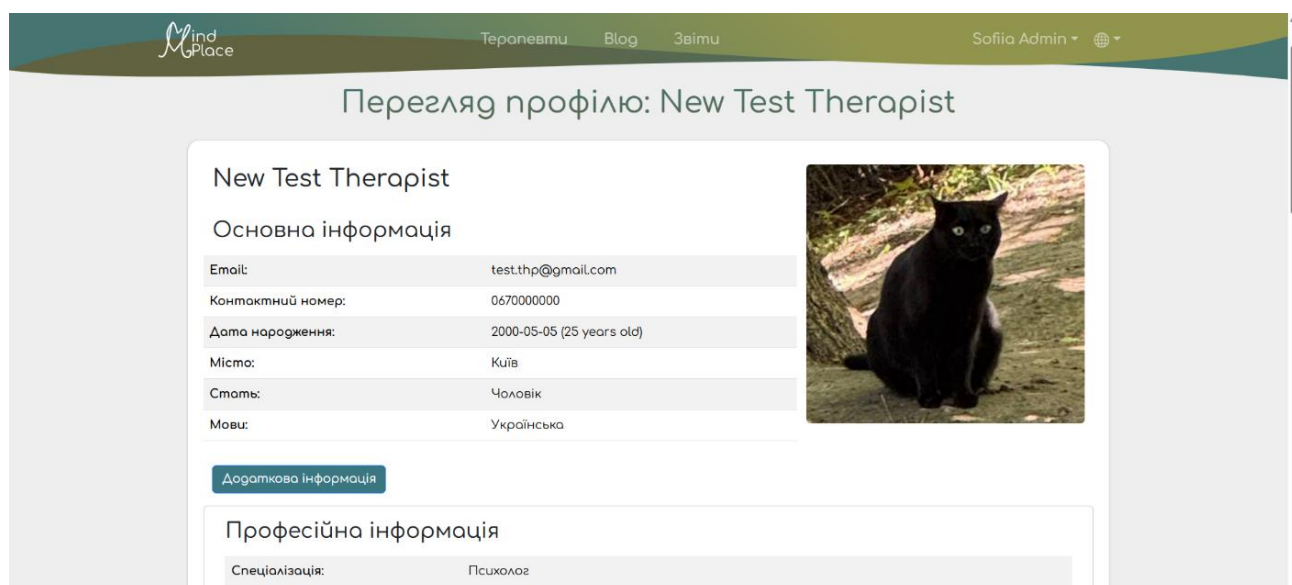


Рис. 3.29. Сторінка адміністраторської панелі

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

Заявки переглядаються на основі основної інформації про персону (Рис. 3.30) та професійної про спеціаліста (Рис. 3.31).

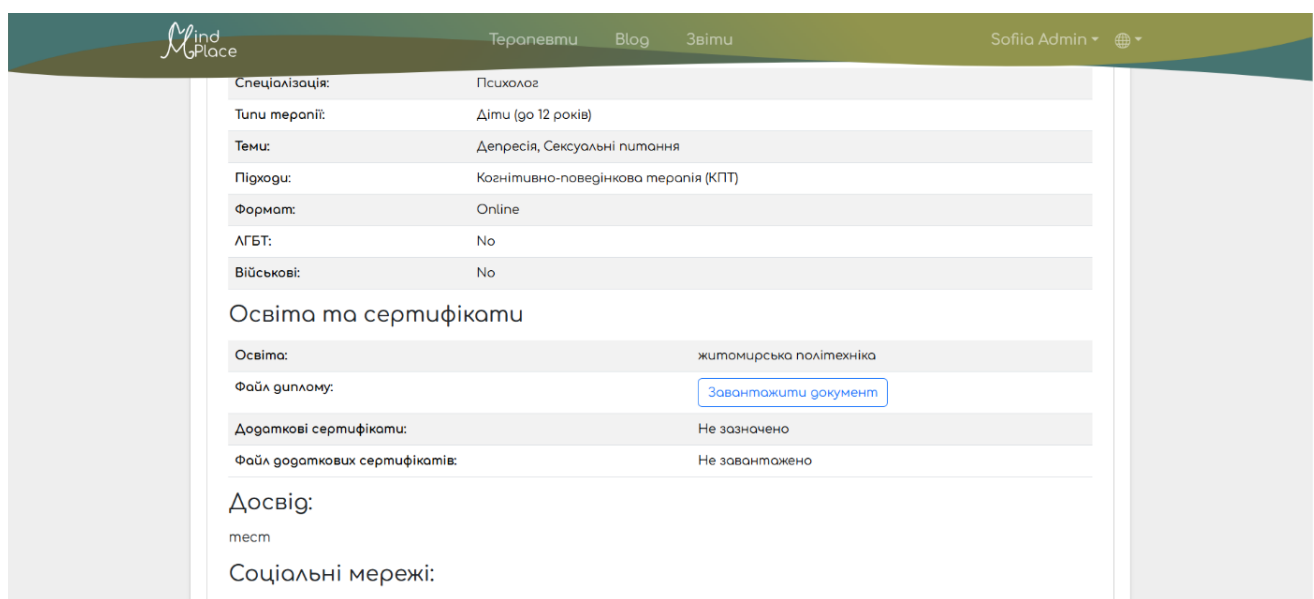


New Test Therapist	
Основна інформація	
Email:	test.thp@gmail.com
Контактний номер:	0670000000
Дата народження:	2000-05-05 (25 years old)
Місто:	Київ
Стать:	Чоловік
Мова:	Українська

Додаткова інформація

Професійна інформація	
Спеціалізація:	Психолог

Рис. 3.30. Перегляд заявки (основна інформація)



Спеціалізація:	Психолог
Типи терапії:	Діти (до 12 років)
Темі:	Депресія, Сексуальні питання
Підходи:	Когнітивно-поведінкова терапія (КПТ)
Формат:	Online
ЛГБТ:	No
Військові:	No

Освіта та сертифікати

Освіта:	житомирська політехніка
Файл диплому:	Завантажити документ
Додаткові сертифікати:	Не зазначено
Файл додаткових сертифікатів:	Не завантажено

Досвід:

тест

Соціальні мережі:

Рис. 3.31. Перегляд заявки (професійна інформація)

Нарешті адміністратор вирішує щодо надання доступу користувачу до ролі лікаря, з можливістю підтвердити або відхилити заявку із зазначенням причини (Рис. 3.32).

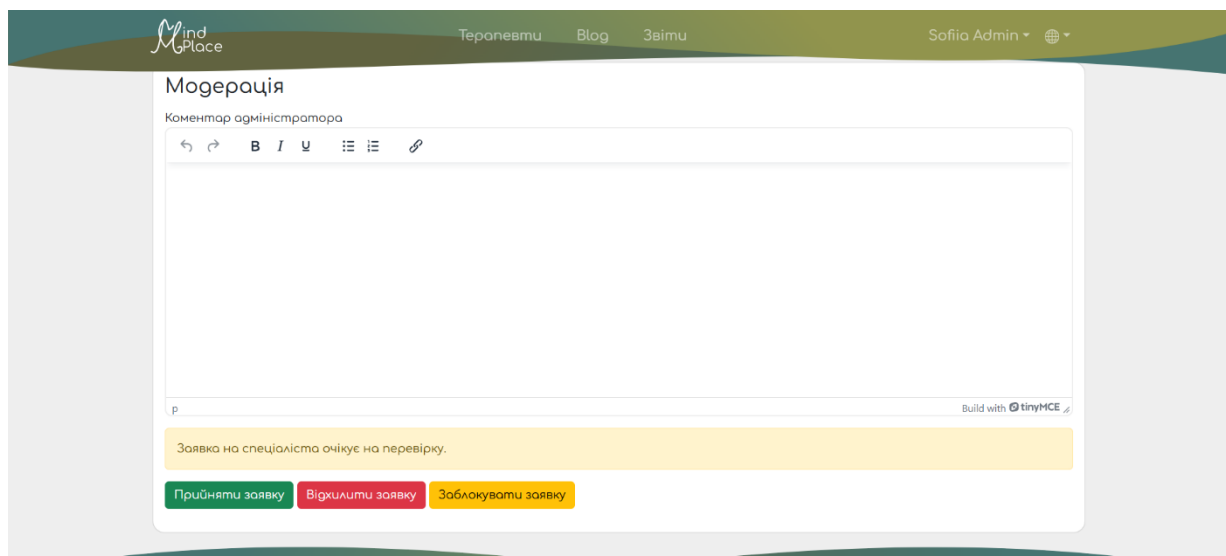


Рис. 3.32. Модерація заявки

Після підтвердження заявки, терапевт отримує доступ до повного функціоналу. Окрім розширеного перегляду профілю (Рис. 3.33), він вже може створювати та керувати власним графіком (Рис. 3.34).

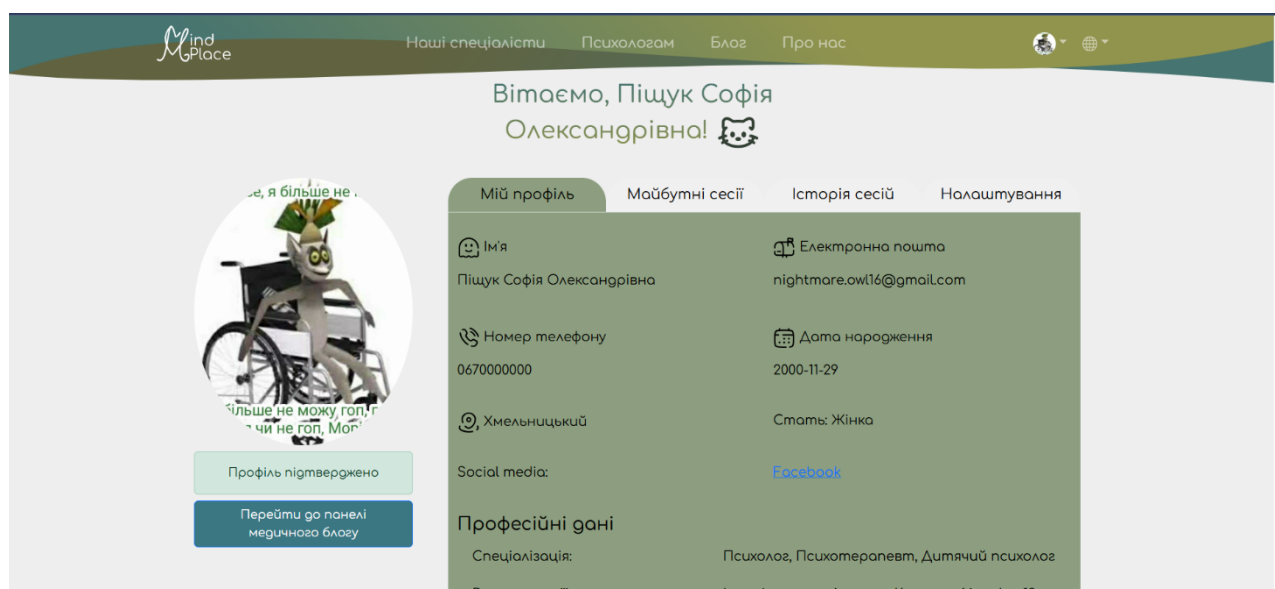


Рис. 3.33. Сторінка підтвердженого спеціаліста (дані профілю)



Рис. 3.34. Сторінка підтвердженого спеціаліста (створити графік)

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

Також може переглядати проведені ним (Рис. 3.36) та майбутні сесії (Рис. 3.35).

Мій профіль Майбутні сесії Історія сесій Налаштування

Зайняті слоти розкладу: [Показати всі](#)

№3 28/54/2025 22:54	Деталі Відмінити
Тривалість: 15 хв. Session: Вільний	
№7 29/55/2025 12:55	Деталі Відмінити
Тривалість: 30 хв. Session: Вільний	
№5 29/55/2025 22:55	Деталі Відмінити
Тривалість: 30 хв. Session: Вільний	
№4 5/55/2025 22:55	Деталі Відмінити
Тривалість: 50 хв. Session: Вільний	

Створити графік:
Дата

Рис. 3.35. Сторінка підтвердженого спеціаліста (майбутні сесії)

Мій профіль Майбутні сесії Історія сесій Налаштування

Дата: 18/00/2025 18:00 Тривалість: 30 хв. Session: Призначено (ID: 11)	Деталі
Дата: 18/54/2025 12:54 Тривалість: 30 хв. Session: Призначено (ID: 10)	Деталі
Дата: 18/54/2025 12:54 Тривалість: 30 хв. Session: Призначено (ID: 10)	Деталі

Рис. 3.36. Сторінка підтвердженого спеціаліста (історія сесій)

Як і звичайному користувачу, йому доступно налаштування особистих даних, але тепер ще з світлиною профілю (Рис. 3.37) та налаштування професійної інформації (Рис. 3.38).

Мій профіль Майбутні сесії Історія сесій Налаштування

Особисті дані

Ваше ім'я: Піщук Софія Олександрівна

Ваш email: nightmare.cwl16@gmail.com

Ваш номер телефону: 0670000000

Date of birth: 29.11.2000

Form photo: [Вибрати файл](#) [Файл не вибрано](#)

Змінити пароль

Введіть пароль: *****

Повторіть пароль: *****

Рис. 3.37. Сторінка підтвердженого спеціаліста (налаштування особистих даних)

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.38. Сторінка підтвердженого спеціаліста (налаштування професійних даних)

Внизу секції вкладки йдуть налаштування документів (Рис. 3.39). Варто підкреслити, що основний документ про освіту вже змінити неможливо, але натомість можна додавати інші сертифікати.

Рис. 3.39. Сторінка підтвердженого спеціаліста (налаштування документів)

На платформі реалізовано розділ блогу (Рис. 3.40) для того щоб поширювати та популяризувати достовірну інформацію щодо ментального здоров'я.

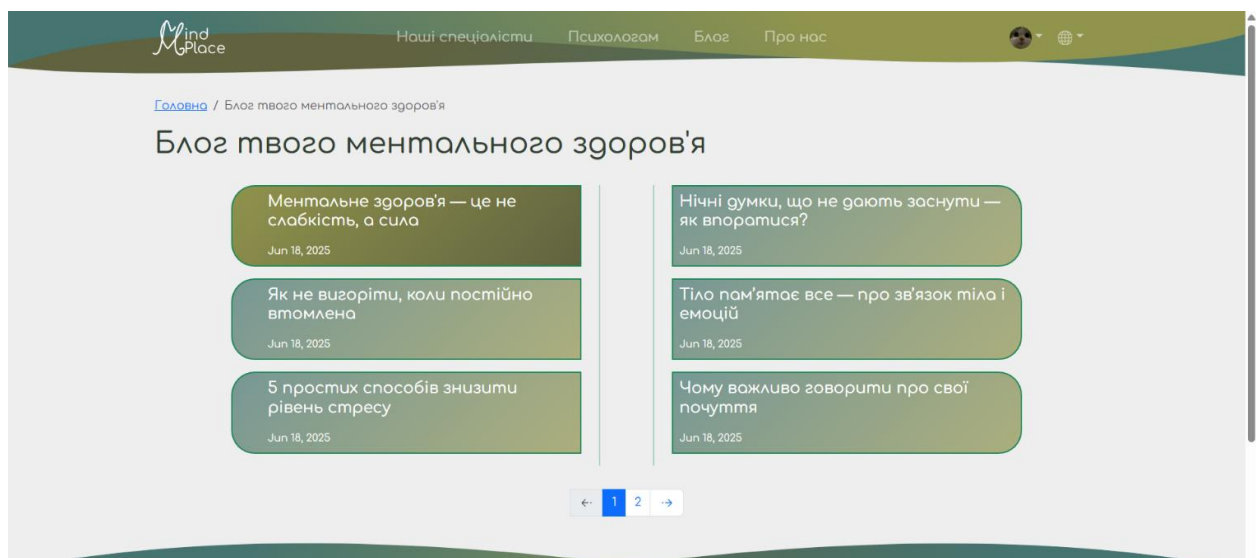


Рис. 3.40. Головна сторінка блогу

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

Натиснувши на заголовок який зацікавив, відкривається сторінка окремої статті (Рис. 3.41). Знизу зазначено автора з посиланням на його сторінку, якщо надана інформація надихне користувача звернутися за допомогою саме до цього спеціаліста.

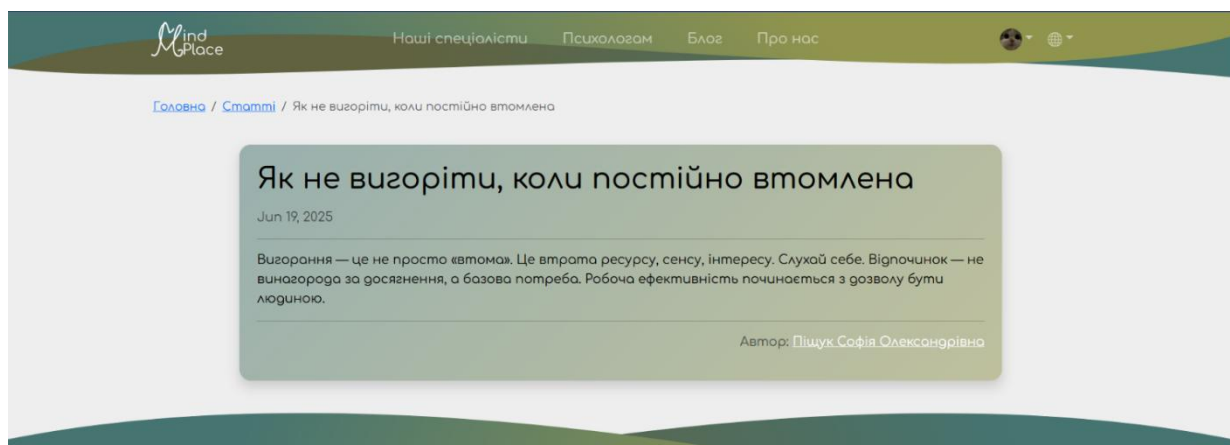


Рис. 3.41. Сторінка статті

Сам спеціаліст може перейти зі сторінки свого профілю до панелі власного медичного блогу (Рис. 3.42). Звідки можна перейти до форми створення нової статті (Рис. 3.43).

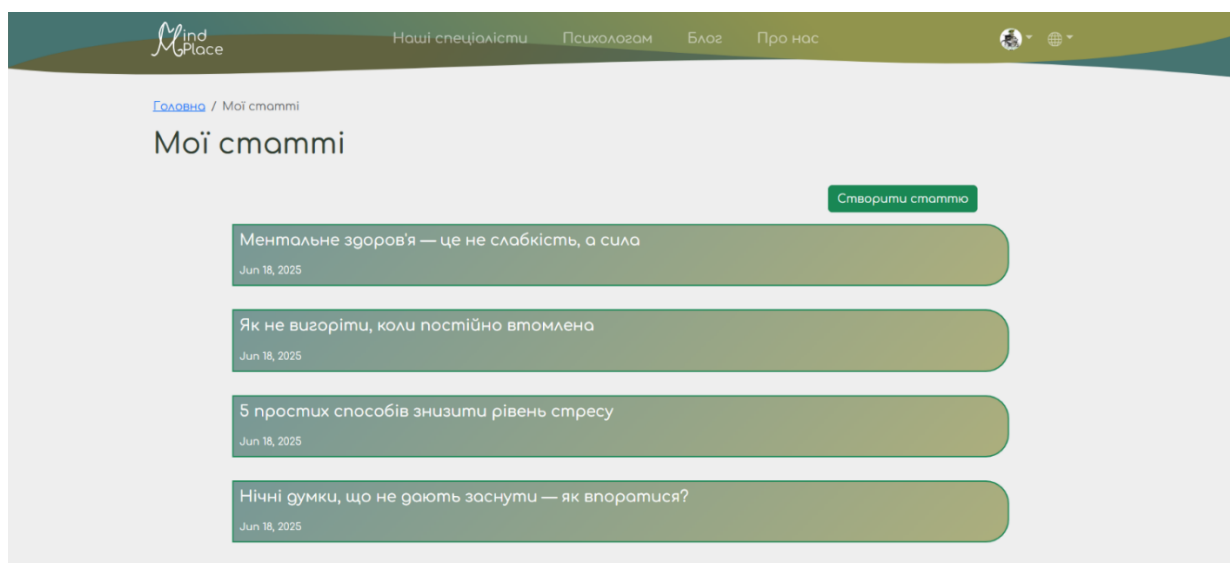


Рис. 3.42. «Мої статті» спеціаліста

Або ж перейти до сторінки конкретної власної статті для маніпуляцій над нею (Рис. 3.44). Також, варто підкреслити, що поле на контенту статті на цій сторінці має розширений функціонал, завдяки чому можна редагувати вміст, вигляд, посилання, фото та багато іншого.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

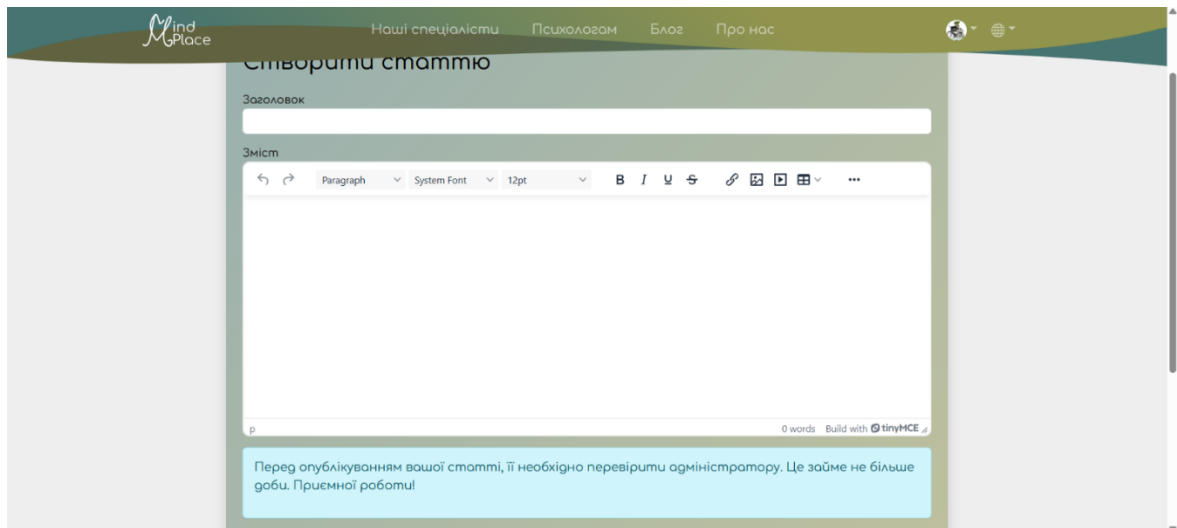


Рис. 3.43. Форма створення нової статті

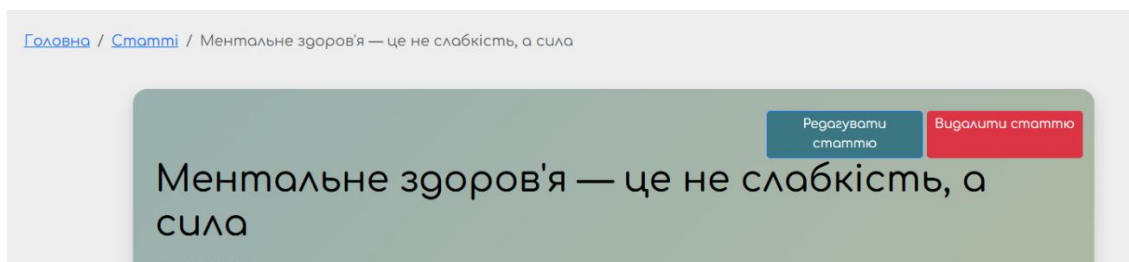


Рис. 3.44. Маніпуляції над власною статтею

Перед опублікуванням, вперше чи повторно, однієї статті її має перевірити адміністратор та допустити до загального перегляду. На адміністраторській панелі, на вкладці «Блог» переглядаються усі існуючі статті за статусом публікації (Рис. 3.45). Як і з заявками спеціалістів, адміністратор може фільтрувати та сортувати записи за потребою.

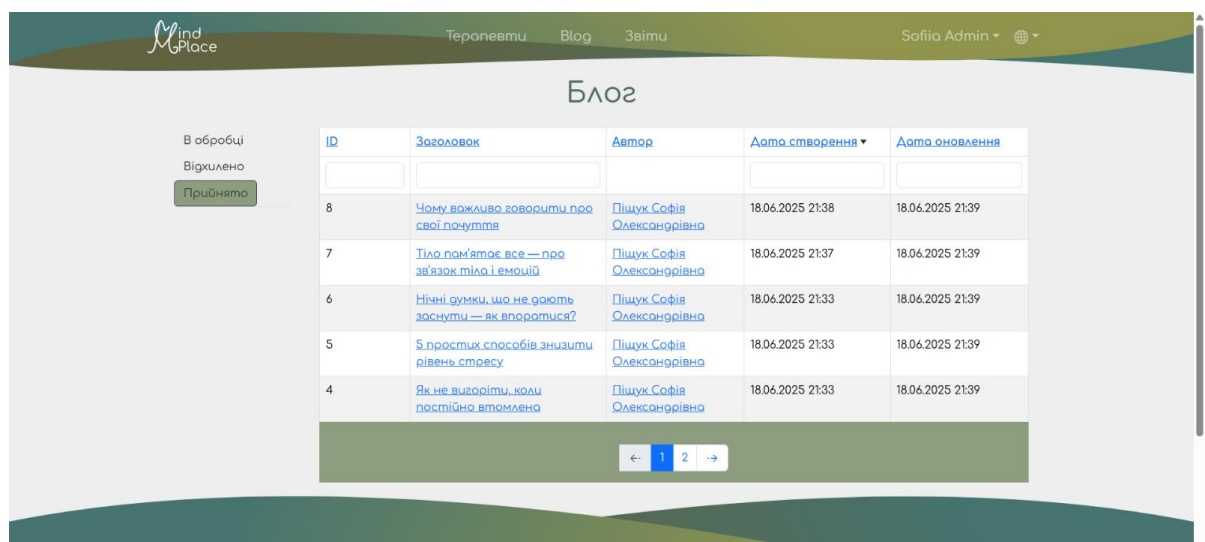


Рис. 3.45. Модерування статтями користувачів

Для того щоб підтвердити публікацію чи відмовити в ній, необхідно перейти на сторінку самої статті (Рис. 3.46). Це гарантує, що адміністратор не прийматиме рішення про допуск тільки по заголовку.

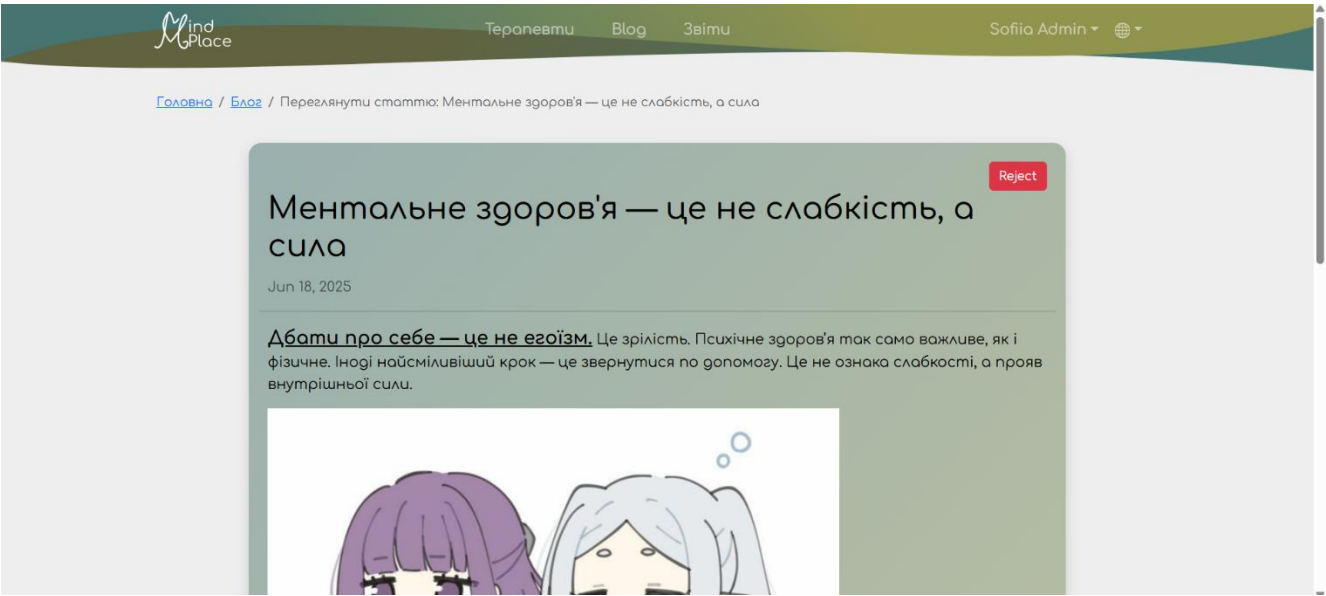


Рис. 3.46. Модерація конкретної статті

За допомогою продуманої структури інтерфейсу, чіткої навігації та гнучкого налаштування ролей користувачів, MindPlace забезпечує ефективну взаємодію між пацієнтами, терапевтами та адміністраторами на всіх етапах користування платформою.

3.3. Тестування роботи програмного комплексу

Тестування розробленої системи MindPlace є важливим етапом життєвого циклу розробки програмного забезпечення, що дозволяє виявити та усунути недоліки до впровадження платформи в реальне середовище. Основну увагу було приділено ручному тестуванню користувацького інтерфейсу, перевірці API-запитів за допомогою Postman [26], а також моделюванню типових сценаріїв подачі заявки та запису на сесію.

У таблиці нижче (табл. 3.1) наведено підсумки тестування чотирьох ключових сценаріїв, що охоплюють основні модулі системи:

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

Результати тестування основних сценаріїв

№	Назва сценарію	Передумови	Кроки	Очікуваний результат	Фактичний результат	Коментар
1	Вхід користувача з підтвердженою роллю	Створено користувача з роллю user	Ввести email/пароль, натиснути «Увійти»	Відкривається головна сторінка з увійденим профілем	Вхід виконано успішно	Немає
2	Надсилання заявки спеціаліста	Користувач авторизований	Перейти на сторінку заявки, пройти всі кроки форми, натиснути «Надіслати»	З'являється повідомлення про успішну подачу заявки	Сценарій працює, дані зберігаються	У першій версії не зберігалися вкладені документи – виправлено
3	Бронювання сесії пацієнтом	Є підтверджений спеціаліст з вільним слотом	Перейти до профілю спеціаліста, обрати слот, підтвердити бронювання	З'являється сторінка з деталями сесії	Працює, запис створюється	Успішно протестувався створення Google Meet
4	Перевірка модерації заявки адміністратором	Надіслана нова заявка	Увійти як адміністратор, переглянути заявку, натиснути «Підтвердити»	Заявка підтверджується, роль змінюється на specialist	Заявка підтверджується користувачу відкриваються нові функції	API-запит PATCH мав неправильну структуру – виправлено

Аналіз результатів тестування показав, що ключові функції платформи працюють відповідно до очікувань, за винятком кількох типових помилок ранньої розробки. Зокрема, виявлено та усунено проблеми з валідацією вкладених форм, некоректною обробкою документів у формі заявки та маршрутизацією запитів до адміністративної панелі.

Особливу увагу було приділено перевірці обміну даними між клієнтською частиною та сервером, правильності обробки сесій користувачів, а також створенню Google Meet-посилань для консультацій. Було підтверджено, що після авторизації користувачі з відповідними ролями отримують доступ лише до дозволених розділів, а система забезпечує належну маршрутизацію відповідно до рівня доступу.

У результаті тестування також підтверджено, що адміністративний модуль модерації заявок дозволяє ефективно переглядати, редагувати та підтверджувати заявки спеціалістів, забезпечуючи прозору взаємодію між лікарями та адміністрацією платформи.

Висновки до третього розділу

У третьому розділі було детально розглянуто структуру, інтерфейс та порядок роботи з вебзастосунком MindPlace, що призначений для підтримки ментального здоров'я користувачів та взаємодії з фахівцями. Розроблена система є повноцінним багаторівневим програмним комплексом, який поєднує сучасні підходи до побудови frontend- і backend-архітектури, підтримує контейнеризацію через Docker і забезпечує гнучку інтеграцію зі сторонніми сервісами, зокрема Google Meet і Amazon S3.

Інтерфейс системи реалізований відповідно до принципів адаптивного дизайну та зручного користувацького досвіду. Завдяки мультирольовій структурі (пацієнт, спеціаліст, адміністратор) платформа задовольняє потреби різних категорій користувачів: від простого перегляду консультацій — до адміністрування заявок, модерування блогу й управління медичною діяльністю.

Особливу увагу приділено реалізації процесу приєднання спеціалістів до платформи через покрокову форму заявки, що містить перевірку персональних, професійних та освітніх даних. Також розроблено розширений кабінет спеціаліста з функціями планування графіку, перегляду майбутніх та минулих консультацій, а також ведення власного блогу. Для пацієнтів забезпечено інструменти пошуку

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

фахівця, запису на консультацію, перегляду історії взаємодій та залишення відгуків.

Проведене тестування підтвердило працездатність ключових сценаріїв системи: вхід користувачів, подача заявки, запис на консультацію, обробка заявок адміністрацією. У процесі тестування було виявлено низку технічних помилок, що притаманні ранній стадії розробки, які були успішно усунені.

Таким чином, система MindPlace реалізована як надійний та масштабований вебзастосунок із зручною структурою, адаптивним інтерфейсом, підтримкою сучасних інтеграцій та широким функціоналом для ефективної співпраці між пацієнтами, спеціалістами та адміністраторами. Результати тестування підтверджують готовність проєкту до практичного впровадження.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				64
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті виконання випускної кваліфікаційної роботи бакалавра було проаналізовано існуючі сервіси у сфері ментального здоров'я, визначено вимоги до функціональності, обґрунтовано вибір технологій і спроектовано архітектуру програмного забезпечення. Аналіз аналогів дозволив виділити ключові потреби цільової аудиторії: безпечний доступ до перевірених спеціалістів, зручне бронювання сесій, конфіденційність комунікації, а також доступ до якісного інформаційного контенту у форматі блогу.

Для розробки системи було обрано архітектурний шаблон MVC, а також PHP-фреймворк Yii2, що обґрунтовано його перевагами. Було складено перелік сучасних інструментів для веброзробки, а для зберігання даних обрана реляційна СУБД MySQL.

Було докладно висвітлено архітектурне й функціональне проєктування додатку для підтримки ментального здоров'я MindPlace. Визначено основні компоненти системи та розкрито принципи їх взаємодії. Розглянуто можливі сценарії використання платформи з урахуванням ролей користувача, адміністратора й спеціаліста, а також типові моделі їх взаємодії із системою.

Проєктований функціонал охоплює надання психологічної підтримки, індивідуальний підбір спеціаліста, створення й перегляд медичних статей, що свідчить про фокус платформи на зручність, доступність та ефективність допомоги в питаннях ментального здоров'я.

Класова діаграма та структура бази даних ілюструють об'єктно-орієнтований підхід до побудови системи, що забезпечує її масштабованість і гнучкість при роботі з основними сутностями. Такий підхід дозволяє чітко організувати логіку взаємодії користувачів, модерацію та управління контентом.

Діаграми активності допомогли деталізувати ключові алгоритми – зокрема, процеси реєстрації, авторизації, процесу отримання та надання консультацій, маніпуляції над контентом. Це забезпечує прозоре розуміння функціонування платформи як для користувачів, так і для розробників.

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

Було детально розглянуто структуру, інтерфейс та порядок роботи з вебзастосунком MindPlace, що призначений для підтримки ментального здоров'я користувачів та взаємодії з фахівцями. Розроблена система є повноцінним багаторівневим програмним комплексом, який поєднує сучасні підходи до побудови frontend- і backend-архітектури, підтримує контейнеризацію через Docker і забезпечує гнучку інтеграцію зі сторонніми сервісами, зокрема Google Meet і Amazon S3.

Інтерфейс системи реалізований відповідно до принципів адаптивного дизайну та зручного користувацького досвіду. Завдяки мультирольовій структурі (пацієнт, спеціаліст, адміністратор) платформа задовольняє потреби різних категорій користувачів: від простого перегляду консультацій — до адміністрування заявок, модерування блогу й управління медичною діяльністю.

Особливу увагу приділено реалізації процесу приєднання спеціалістів до платформи через покрокову форму заявки, що містить перевірку персональних, професійних та освітніх даних. Також розроблено розширений кабінет спеціаліста з функціями планування графіку, перегляду майбутніх та минулих консультацій, а також ведення власного блогу. Для пацієнтів забезпечено інструменти пошуку фахівця, запису на консультацію, перегляду історії взаємодій та залишення відгуків.

Проведене тестування підтвердило працездатність ключових сценаріїв системи: вхід користувачів, подача заявки, запис на консультацію, обробка заявок адміністрацією. У процесі тестування було виявлено низку технічних помилок, що притаманні ранній стадії розробки, які були успішно усунені.

Таким чином, додаток MindPlace реалізовано як надійний та масштабований вебзастосунок із зручною структурою, адаптивним інтерфейсом, підтримкою сучасних інтеграцій та широким функціоналом для ефективної співпраці між пацієнтами, спеціалістами та адміністраторами. Результати тестування підтверджують готовність проєкту до практичного впровадження.

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Google Meet [Електронний ресурс] – Режим доступу до ресурсу:
<https://meet.google.com/landing>
2. Google Calendar [Електронний ресурс] – Режим доступу до ресурсу:
<https://calendar.google.com/>
3. Mentoly [Електронний ресурс] – Режим доступу до ресурсу:
<https://mentoly.com.ua>
4. Розкажи мені (Tell me) [Електронний ресурс] – Режим доступу до ресурсу:
<https://tellme.com.ua/>
5. Mindly [Електронний ресурс] – Режим доступу до ресурсу:
<https://mindlyspace.com/>
6. MVC Design Pattern [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.geeksforgeeks.org/system-design/mvc-design-pattern/>
7. Docker Documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.docker.com/>
8. Yii2 [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.yiiframework.com/doc/guide/2.0/uk>
9. MySQL [Електронний ресурс] – Режим доступу до ресурсу:
<https://dev.mysql.com/doc/>
10. Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites / Robin Nixon – [6 ed.] – O'Reilly Media, 2021. – 823 p.
11. PHP Manual [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.php.net/manual/uk/>
12. RESTful [Електронний ресурс] – Режим доступу до ресурсу:
<https://yiiframework.com.ua/uk/doc/guide/2/rest-quick-start/>

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижевотря О.В.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

13. Css [Електронний ресурс] – Режим доступу до ресурсу:
<https://developer.mozilla.org/en-US/docs/Web/CSS>
14. Bootstrap Documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
15. Sass: Syntactically Awesome Style Sheets [Електронний ресурс] – Режим доступу до ресурсу: <https://sass-lang.com/>
16. PlantText Diagrams [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.planttext.com/>
17. UML Activity Diagram [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.lucidchart.com/pages/uml-activity-diagram>
18. UML Class Diagrams [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>
19. Working with JSON [Електронний ресурс] – Режим доступу до ресурсу:
https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/JSON
20. Ajax jQuery API Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://api.jquery.com/category/ajax/>
21. Amazon S3 [Електронний ресурс] – Режим доступу до ресурсу:
<https://aws.amazon.com/s3/>
22. Google Cloud Platform [Електронний ресурс] – Режим доступу до ресурсу:
<https://console.cloud.google.com/>
23. Git Documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://git-scm.com/docs>
24. NGINX Docker Official Image [Електронний ресурс] – Режим доступу до ресурсу: https://hub.docker.com/_/nginx

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

25. TinyMCE [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.tiny.cloud/>
26. Postman Docs [Електронний ресурс] – Режим доступу до ресурсу:
<https://learning.postman.com/>
27. JavaScript Guide [Електронний ресурс] – Режим доступу до ресурсу:
<https://javascript.info/>
28. People API [Електронний ресурс] – Режим доступу до ресурсу:
<https://console.cloud.google.com/apis/api/people.googleapis.com/>
29. Core Validators [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.yiiframework.com/doc/guide/2.0/en/tutorial-core-validators>
30. OAuth 2.0 [Електронний ресурс] – Режим доступу до ресурсу:
<https://auth0.com/>

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

Додатки

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг DockerFile:

```

FROM php:8.4-fpm

RUN apt-get update && apt-get install -y \
    git unzip libpq-dev libpng-dev libjpeg-dev libfreetype6-dev \
    && docker-php-ext-configure gd --with-freetype --with-jpeg \
    && docker-php-ext-install pdo pdo_mysql gd

RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --
filename=composer

RUN apt-get update && apt-get install -y cron

COPY ./crontab /etc/cron.d/session-cron

RUN chmod 0644 /etc/cron.d/session-cron && \
    crontab /etc/cron.d/session-cron

RUN docker-php-ext-install pdo pdo_mysql

WORKDIR /var/www/html

```

Лістинг docker-compose.yml:

```

# version: '3.8'

services:
  php:
    build: ./docker/php
    container_name: yii2-php
    volumes:
      - ./src:/var/www/html
      - ~/.composer-docker/cache:/root/.composer/cache:delegated
      - ./src/web/assets:/var/www/html/web/assets
      - ./src/runtime:/app/runtime
    networks:
      - yii2-network
    depends_on:
      - mysql

  nginx:
    image: nginx:latest
    container_name: yii2-nginx
    volumes:
      - ./src:/var/www/html
      - ./docker/nginx/default.conf:/etc/nginx/conf.d/default.conf
    ports:
      - "8080:80"
    networks:
      - yii2-network
    depends_on:
      - php

  mysql:

```

```

image: mysql:8.0
container_name: yii2-mysql
restart: always
environment:
  MYSQL_ROOT_PASSWORD: root
  MYSQL_DATABASE: yii2
  MYSQL_USER: sofiaa
  MYSQL_PASSWORD: root
volumes:
  - ./docker/mysql:/etc/mysql/conf.d
  - mysql_data:/var/lib/mysql
ports:
  - "3306:3306"
networks:
  - yii2-network
sass:
  image: node:22
  working_dir: /app
  volumes:
    - ./app
  command: npm run sass
volumes:
  mysql_data:
networks:
  yii2-network:

```

Лістинг User.php:

```

<?php

namespace app\models;

use app\services\UserAuthService;
use app\models\forms\FormOptions;

use Yii;
use yii\web\IdentityInterface;
use yii\db\ActiveRecord;

/**
 * User model
 *
 * @property int $id
 * @property string $name
 * @property string $email
 * @property string $contact_number (optional)
 * @property string $date_of_birth
 * @property string $password_hash
 * @property string $role
 * @property string $auth_key
 * @property string|null $google_token
 * @property string $access_token
 * @property string $created_at
 * @property string $auth_type
 * @property string|null $city
 * @property string|null $address
 * @property string|null $gender

```

		Піщук С.О.			Житомирська політехніка. 25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				72
Змн.	Арк.	№ докум.	Підпис	Дата		

```

* @property string[]|null $language
* @property string[]|null $therapy_types JSON-масив
* @property string[]|null $theme JSON-масив
* @property string[]|null $approach_type JSON-масив
* @property string[]|null $format JSON-масив
* @property bool|null $lgbt
* @property bool|null $military
* @property string[]|null $specialization JSON-масив
* @property string|null $education_name
* @property string|null $education_file
* @property string|null $education_file_url
* @property string|null $additional_certification
* @property string|null $additional_certification_file
* @property string|null $additional_certification_file_url
* @property string[]|null $experience JSON-масив
* @property string[]|null $social_media JSON-масив
* @property string|null $photo
* @property string|null $photo_url
*/
class User extends ActiveRecord implements IdentityInterface
{
    /**
     * Table name in the database
     *
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'user';
    }

    #region CRUD

    /**
     * Generates a new user
     *
     * @param array $data user data
     * @return User|null created user or null if failed
     */
    public static function createUser($data)
    {
        $user = new self();
        if (!$user->validate()) {
            Yii::error($user->getErrors(), 'user-create');
            return null;
        }

        // Set default user properties
        $user->name = $data['name'] ?? null;
        $user->email = $data['email'] ?? null;
        $user->contact_number = $data['contact_number'] ?? null;
        $user->date_of_birth = $data['date_of_birth'] ?? null;
        $user->role = $data['role'] ?? 'default';

        // Set security properties
        $user->password_hash = UserAuthService::hashPassword($data['password']);
        $user->auth_key = Yii::$app->security->generateRandomString();
        $user->access_token = Yii::$app->security->generateRandomString(255);
    }
}

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $user->save() ? $user : null;
    }

    /**
     * Updates user information
     *
     * @param int $id user ID
     * @param array $data user data
     * @return string|false access_token або false у разі помилки
     */
    public static function updateUser($id, $data)
    {
        $user = self::findOne($id);
        Yii::info('User update data: ' . json_encode($data), 'user-update');

        if (!$user) {
            Yii::error("User ID $id not found", 'user-update');
            return false;
        }

        if (isset($data['email'])) {
            $existing = self::findByEmail($data['email']);
            if ($existing && $existing->id !== $id) {
                Yii::error("Email {$data['email']} is already taken by another user",
                'user-update');
                return false;
            }
        }

        $jsonFields = [
            'language',
            'therapy_types',
            'theme',
            'approach_type',
            'format',
            'specialization',
            'experience',
            'social_media'
        ];

        foreach ($data as $field => $value) {
            if (in_array($field, $jsonFields, true)) {
                $user->$field = is_array($value) ? json_encode($value,
                JSON_UNESCAPED_UNICODE) : $value;
            } else {
                Yii::info("$field = $value", 'user-update');
                $user->$field = $value;
            }
        }

        foreach (['education', 'additional_certification'] as $prefix) {
            if (!empty($data["{$prefix}_file"])) {
                $user->{"{$prefix}_file"} = $data["{$prefix}_file"];
                $user->{"{$prefix}_file_url"} = $data["{$prefix}_file_url"] ?? null;
            }
        }

        if (!empty($data["photo"])) {
            $user->photo = $data["photo"];
            $user->photo_url = $data["photo_url"] ?? null;
        }
    }

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (!empty($data['password']) && !empty($data['re_password'])) {
            if ($data['password'] !== $data['re_password']) {
                Yii::warning('Passwords do not match', 'user-update');
                return false;
            }
            $user->password_hash = UserAuthService::hashPassword($data['password']);
        }

        if (!$user->save()) {
            Yii::error('User save failed: ' . json_encode($user->getErrors()), 'user-
update');
            throw new \RuntimeException('Помилка збереження користувача.');
```

```

            return false;
        }

        return $user->access_token;
    }

    /**
     * Оновлення налаштувань користувача для поточного користувача
     *
     * @param array $data user data
     * @return string|false access_token або false у разі помилки
     */
    public static function updateUserSettings(array $data)
    {
        if (empty($data)) {
            Yii::error('Дані для оновлення порожні або не є масивом', 'user-update-
settings');
            Yii::$app->session->setFlash('error', 'Помилка при оновленні профілю');
            return false;
        }

        $user = Yii::$app->user->identity;
        if (!$user) {
            Yii::error('Користувач не авторизований', 'user-update-settings');
            Yii::$app->session->setFlash('error', 'Помилка при оновленні профілю');
            return false;
        }

        return self::updateUser($user->id, $data);
    }

    /**
     * Deletes a user
     *
     * @param int $id user ID
     * @return bool whether the deletion was successful
     */
    public static function deleteUser($id)
    {
        $user = self::findOne($id);
        return $user ? (bool) $user->delete() : false;
    }

    /**

```

```

* Finds user by name
*
* @param string $name
* @return static|null
*/
public static function findByName($name)
{
    return self::findOne(['name' => $name]);
}

/**
* Finds user by id
*
* @param int $id
* @return static|null
*/
public static function findById($id)
{
    return self::findOne(['id' => $id]);
}

/**
* Finds user by email
*
* @param string $email
* */
public static function findByEmail($email)
{
    return self::findOne(['email' => $email]);
}

public static function getSpecialists($params = '')
{
    return self::find()->where(['role' => 'specialist'])->all();
}

public static function getPatients($params = [])
{
    return self::find()->where(['role' => 'default'])->all();
}

public static function getSpecializationList($params = '')
{
    return [
        'psychotherapist' => 'Psychotherapist',
        'psychologist' => 'Psychologist',
    ];
}

public static function getStatusList($params = '')
{
    return [
        'pending' => 'Pending',
        'approved' => 'Approved',
        'rejected' => 'Rejected',
        'blocked' => 'Blocked',
    ];
}
}
#endregion

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

```

#region IdentityInterface

/**
 * Finds user by ID
 *
 * {@inheritdoc}
 */
public static function findIdentity($id)
{
    $user = self::findOne($id);
    if ($user === null) {
        Yii::error("findIdentity: User not found for ID = $id", __METHOD__);
    }
    return $user;
}

/**
 * Finds user by access token
 *
 * {@inheritdoc}
 */
public static function findIdentityByAccessToken($token, $type = null)
{
    return self::findOne(['access_token' => $token]);
}

/**
 * Gets the ID of the user
 *
 * {@inheritdoc}
 */
public function getId()
{
    return $this->id;
}

/**
 * Gets the auth key of the user
 *
 * {@inheritdoc}
 */
public function getAuthKey()
{
    return $this->auth_key;
}

/**
 * Validates the auth key
 *
 * {@inheritdoc}
 */
public function validateAuthKey($authKey)
{
    return $this->auth_key === $authKey;
}

public function validatePassword($password)
{
    return Yii::$app->security->validatePassword($password, $this->password_hash);
}

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

#endregion

#region Roles & Access Control

/**
 * Checks if the user has the specified role
 *
 * @param string $role role to check
 * @return bool whether the user has the specified role
 */
public function hasRole($role)
{
    return $this->role === $role;
}

/**
 * Checks if the user is an admin
 * Admin has the highest level of access and can perform all actions
 * including managing users, articles, and managing specialists
 *
 * @return bool whether the user is an admin
 */
public function isAdmin()
{
    return $this->role === 'admin';
}

/**
 * Checks if the user is a moderator
 * Moderator has a lower level of access than admin and can perform some actions
 * like managing articles, reviews and reports
 *
 * @return bool whether the user is a moderator
 */
public function isModerator()
{
    return $this->role === 'moderator';
}

/**
 * Checks if the user is a specialist (doctor)
 * Specialist has a lower level of access than moderator and can perform some actions
 * 1. creating articles
 * 2. edit their own articles
 * 3. edit own profile
 * 4. chat with users
 *
 * @return bool whether the user is a specialist
 */
public function isSpecialist()
{
    return $this->role === 'specialist';
}

/**
 * Checks if the user is a regular user
 * Regular user has the lowest level of access and can perform some actions
 * 1. edit own profile

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

* 2. chat with specialists
* 3. leave reviews
* 4. leave reports
* 5. view articles
* 6. view specialists
*
* @return bool whether the user is a regular user
*/
public function isUser()
{
    return $this->role === 'default';
}

/**
 * Checks if the user is a guest
 * Guest has no access to any actions and can only view articles and specialists
 *
 * @return bool whether the user is a guest
 */
public function isGuest()
{
    return $this->role === 'guest';
}

/**
 * Checks if the user is logged in
 *
 * @return bool whether the user is logged in
 */
public function isLoggedIn()
{
    return !Yii::$app->user->isGuest;
}
#endregion

public function getAge(): int
{
    if (!$this->date_of_birth) {
        return 0;
    }
    $dob = new \DateTime($this->date_of_birth);
    $today = new \DateTime();
    return $today->diff($dob)->y;
}

public function getTherapyTypesAsArray(): array
{
    return $this->getDecodedAttribute('therapy_types');
}

public function getThemeAsArray(): array
{
    return $this->getDecodedAttribute('theme');
}

public function getApproachTypeAsArray(): array
{
    return $this->getDecodedAttribute('approach_type');
}

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				79
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    public function getFormatAsArray(): array
    {
        return $this->getDecodedAttribute('format');
    }

    public function getSpecializationAsArray(): array
    {
        return $this->getDecodedAttribute('specialization');
    }

    public function getExperienceAsArray(): array
    {
        return $this->getDecodedAttribute('experience');
    }

    public function getSocialMediaAsArray(): array
    {
        return $this->getDecodedAttribute('social_media');
    }

    public function getLanguageAsArray(): array
    {
        return $this->getDecodedAttribute('language');
    }

    public function getOptionLabel(string $fieldName, string $optionCategory): ?string
    {
        $value = $this->$fieldName;
        return FormOptions::getLabel($optionCategory, $value);
    }

    protected function normalizeJsonArray($value): array
    {
        if (is_array($value)) return $value;
        if (is_string($value)) {
            $decoded = json_decode($value, true);
            return is_array($decoded) ? $decoded : [];
        }
        return [];
    }

    public function getOptionLabels(string $fieldName, string $optionCategory): array
    {
        $keys = $this->getDecodedAttribute($fieldName);
        return array_filter(array_map(fn($k) => FormOptions::getLabel($optionCategory,
$k), $keys));
    }

    public function getDecodedAttribute(string $attr): array
    {
        return $this->normalizeJsonArray($this->$attr ?? []);
    }
}

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг SpecialistApplication.php:

```
<?php

namespace app\models;

use app\services\UserAuthService;
use Yii;

/**
 * SpecialistApplication model
 *
 * @property int $id
 * @property string $user_id
 * @property string $admin_id (optional)
 * @property string $status
 * @property string $comment (optional)
 * @property string $created_at
 * @property string $updated_at
 */
class SpecialistApplication extends User
{
    public const STATUS_PENDING = 'pending';
    public const STATUS_APPROVED = 'approved';
    public const STATUS_REJECTED = 'rejected';
    public const STATUS_BLOCKED = 'blocked';

    public static function tableName()
    {
        return 'specialist_application';
    }

    public function rules()
    {
        return [
            [['comment', 'status'], 'string'],
            [['user_id'], 'required'],
        ];
    }

    public static function updateUserFromTherapistForm($userId, $formData)
    {
        $transaction = Yii::$app->db->beginTransaction();
        try {
            if (!User::updateUser($userId, $formData)) {
                Yii::error('Failed to update user', 'specialist-application');
                return false;
            }

            $SpecialistApplication = Yii::$app->db->createCommand('SELECT id FROM
            {%specialist_application} WHERE user_id = :userId')
                ->bindValue(':userId', $userId)
                ->queryScalar();

            if (!$SpecialistApplication) {
                // new record
                Yii::$app->db->createCommand()->insert('{{%specialist_application}}', [
                    'user_id' => $userId,
                    'status' => 'pending',
                    'created_at' => new \yii\db\Expression('NOW()'),
                ]);
            }
        } catch (\Exception $e) {
            $transaction->rollBack();
            throw $e;
        }
        $transaction->commit();
    }
}
```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        ]->execute();
    } else {
        // update existing record
        Yii::$app->db->createCommand()->update('{{%specialist_application}}', [
            'status' => 'pending',
            'updated_at' => new \yii\db\Expression('NOW()'),
        ], 'user_id = :userId', [':userId' => $userId])->execute();
    }

    $transaction->commit();
    Yii::info('Successfully saved', 'therapist-join');
    return true;
} catch (\Exception $e) {
    $transaction->rollBack();
    Yii::error('Failed to save SpecialistApplication: ' . $e->getMessage(),
'therapist-join');
    return false;
}
}

public static function updateStatus($status)
{
    $transaction = Yii::$app->db->beginTransaction();
    try {
        $specialistApplications = self::find()->all();
        foreach ($specialistApplications as $specialistApplication) {
            $userId = $specialistApplication->user_id;
            $user = User::findOne($userId);
            if ($user) {
                $user->role = 'specialist';
                $user->save();
                $specialistApplication->status = self::STATUS_APPROVED;
                $specialistApplication->save();
            }
        }
    } catch (\Exception $e) {
        $transaction->rollBack();
        Yii::error('Failed to save SpecialistApplication: ' . $e->getMessage(),
'therapist-join');
        return false;
    }
}

public static function deleteSpecialistApplication($id)
{
    $specialistApplication = self::findOne($id);
    return $specialistApplication ? (bool)$specialistApplication->delete() : false;
}

public static function getStatusById($userId)
{
    $specialistApplication = self::findOne(['user_id' => $userId]);
    return $specialistApplication?->status;
}

public static function getByUserId($userId)
{
    $specialistApplication = self::findOne(['user_id' => $userId]);
    return $specialistApplication ? $specialistApplication : null;
}

public function getUser()
{

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $this->hasOne(User::class, ['id' => 'user_id']);
    }

    public static function getAllByStatus($status)
    {
        return self::find()->where(['status' => $status]);
    }
}

```

Лістинг Schedule.php:

```

<?php

namespace app\models;

use Yii;
use yii\db\ActiveRecord;

/**
 * @property int $id
 * @property int $doctor_id
 * @property string $datetime
 * @property int $duration
 * @property string|null $details
 * @property int|null $client_id
 * @property string|null $status
 * @property string|null $meet_url
 * @property string|null $recommendations
 */
class Schedule extends ActiveRecord
{
    public const STATUS_SCHEDULED = 'scheduled';
    public const STATUS_BOOKED = 'booked';
    public const STATUS_COMPLETED = 'completed';
    public const STATUS_CANCELED = 'cancelled';

    public static function getFutureTimeCondition(): array
    {
        return ['>', 'datetime', date('Y-m-d H:i:s')];
    }

    public static function getPastTimeCondition(): array
    {
        return ['<', 'datetime', date('Y-m-d H:i:s')];
    }

    public static function tableName()
    {
        return 'schedule';
    }

    public function rules()
    {
        return [
            [['doctor_id', 'datetime', 'duration'], 'required'],
            [['doctor_id', 'duration', 'client_id'], 'integer'],
            [['datetime'], 'datetime', 'format' => 'php:Y-m-d\TH:i'],
            [['status'], 'string', 'max' => 50],
            [['meet_url'], 'string', 'max' => 255],
        ];
    }
}

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				83
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        [['recommendations'], 'string'],
        ['details', 'safe'],
        [['doctor_id'], 'exist', 'skipOnError' => true, 'targetClass' => User::class,
'targetAttribute' => ['doctor_id' => 'id']],
        [['client_id'], 'exist', 'skipOnError' => true, 'targetClass' => User::class,
'targetAttribute' => ['client_id' => 'id']],
    ];
}

public function beforeValidate()
{
    $this->datetime = date('Y-m-d\TH:i', strtotime($this->datetime));
    return parent::beforeValidate();
}

public static function createSchedule($data)
{
    $schedule = new Schedule();
    if (!$schedule->validate()) {
        Yii::error($schedule->getErrors(), 'session-create');
        return null;
    }
    $schedule->doctor_id = $data['doctor_id'] ?? null;
    $schedule->datetime = $data['datetime'] ?? null;
    $schedule->duration = $data['duration'] ?? null;
    $schedule->client_id = $data['client_id'] ?? null;
    $schedule->details = $data['details'] ?? null;
    $schedule->status = self::STATUS_SCHEDULED;
    $schedule->meet_url = $data['meetUrl'] ?? null;
    $schedule->recommendations = $data['recommendations'] ?? null;
    if (!$schedule->save()) {
        Yii::error('Failed to create session' . json_encode($schedule->getErrors()),
'session-create');
        Yii::$app->session->setFlash('error', 'Failed to create schedule. Please try
again later.');
```

```

        return null;
    } else {
        Yii::$app->session->setFlash('success', 'Schedule created successfully.');
```

```

        return $schedule;
    }
}

public static function updateSchedule($id, $data)
{
    $schedule = self::getScheduleById($id);
    if (!$schedule) {
        return null;
    }
    if (!$schedule->validate()) {
        Yii::error($schedule->getErrors(), 'session-update');
        return null;
    }
    $schedule->doctor_id = $data['doctor_id'] ?? $schedule->doctor_id;
    $schedule->client_id = $data['client_id'] ?? $schedule->client_id;
    $schedule->status = $data['status'] ?? $schedule->status;
    $schedule->datetime = $data['datetime'] ?? $schedule->datetime;
    $schedule->duration = $data['duration'] ?? $schedule->duration;
    $schedule->meet_url = $data['meetUrl'] ?? $schedule->meet_url;
    $schedule->details = $data['details'] ?? $schedule->details;
}
```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				84
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $schedule->recommendations = $data['recommendations'] ?? $schedule-
>recommendations;
        Yii::info($schedule->getAttributes(), 'session-update');
        if (!$schedule->save()) {
            Yii::error('Failed to update session' . json_encode($schedule->getErrors()),
'session-update');
            return null;
        } else {
            return $schedule;
        }
    }

    public static function updateMeetLink($id, $link)
    {
        $schedule = self::getScheduleById($id);
        if (!$schedule) {
            return null;
        }
        $schedule->meet_url = $link;
        if (!$schedule->save()) {
            Yii::error('Failed to update session link' . json_encode($schedule-
>getErrors()), 'session-update');
            return null;
        }
        return $schedule;
    }

    public static function deleteSchedule($id)
    {
        $schedule = self::getScheduleById($id);
        if (!$schedule) {
            return null;
        }
        if (!$schedule->delete()) {
            Yii::error('Failed to delete session' . json_encode($schedule->getErrors()),
'session-delete');
            return null;
        } else {
            Yii::$app->session->setFlash('success', 'Schedule deleted successfully.');
```

```

    }
    if ($timeCondition) {
        $schedulesQuery->andWhere([$timeCondition[0], 'datetime', $timeCondition[2] ??
$timeCondition[1]]);
    }
    switch ($timeCondition[0]) {
        case '>=':
            $schedulesQuery->orderBy(['datetime' => SORT_ASC]);
            break;
        case '<':
            $schedulesQuery->orderBy(['datetime' => SORT_DESC]);
            break;
        default:
            $schedulesQuery->orderBy(['datetime' => SORT_ASC]);
            break;
    }
    return $schedulesQuery;
}

public static function getSchedulesByClientId($clientId, $timeCondition = null)
{
    $schedulesQuery = self::find()
        ->where(['client_id' => $clientId]);
    if ($timeCondition) {
        $schedulesQuery->andWhere([$timeCondition[0], 'datetime', $timeCondition[2] ??
$timeCondition[1]]);
    }
    switch ($timeCondition[0]) {
        case '>=':
            $schedulesQuery->orderBy(['datetime' => SORT_ASC]);
            break;
        case '<':
            $schedulesQuery->orderBy(['datetime' => SORT_DESC]);
            break;
        default:
            $schedulesQuery->orderBy(['datetime' => SORT_ASC]);
            break;
    }
    return $schedulesQuery;
}

public static function getScheduleById($id)
{
    return self::findOne(['id' => $id]);
}

public function getDoctor()
{
    return $this->hasOne(User::class, ['id' => 'doctor_id']);
}

public function getClient()
{
    return $this->hasOne(User::class, ['id' => 'client_id']);
}

public function getEndTime()
{
    $start = new \DateTime($this->datetime, new \DateTimeZone(Yii::$app->timeZone));
    $start->add(new \DateInterval('PT' . $this->duration . 'M'));
}

```

```

        return $start->format('H:i');
    }

    public function isBooked(): bool
    {
        return !is_null($this->client_id);
    }

    public function getTherapyTypes()
    {
        return $this->details['therapy_types'] ?? null;
    }

    public function setTherapyTypes($value)
    {
        $details = $this->details ?? [];
        $details['therapy_types'] = $value;
        $this->details = $details;
    }

    public function getTheme()
    {
        return $this->details['theme'] ?? null;
    }

    public function setTheme($value)
    {
        $details = $this->details ?? [];
        $details['theme'] = $value;
        $this->details = $details;
    }

    public function getApproachType()
    {
        return $this->details['approach_type'] ?? null;
    }

    public function setApproachType($value)
    {
        $details = $this->details ?? [];
        $details['approach_type'] = $value;
        $this->details = $details;
    }

    public function getFormat()
    {
        return $this->details['format'] ?? null;
    }

    public function setFormat($value)
    {
        $details = $this->details ?? [];
        $details['format'] = $value;
        $this->details = $details;
    }

    public function getComment()
    {
        return $this->details['comment'] ?? null;
    }

    public function setComment($value)

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				87
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    {
        $details = $this->details ?? [];
        $details['comment'] = $value;
        $this->details = $details;
    }
}

```

Лістинг Review.php:

```

<?php

namespace app\models;

use Yii;
use yii\db\ActiveRecord;

/**
 * This is the model class for table "review".
 *
 * @property int $id
 * @property int $doctor_id
 * @property int $client_id
 * @property int $rating
 * @property string|null $comment
 * @property string $date
 *
 * @property User $doctor
 * @property User $client
 */
class Review extends ActiveRecord
{
    public static function tableName(): string
    {
        return 'review';
    }

    public function rules(): array
    {
        return [
            [['doctor_id', 'client_id', 'rating'], 'required'],
            [['doctor_id', 'client_id', 'rating'], 'integer'],
            [['rating'], 'in', 'range' => [1, 2, 3, 4, 5]],
            [['comment'], 'string'],
            [['date'], 'safe'],

            [['doctor_id'], 'exist', 'skipOnError' => true, 'targetClass' => User::class,
'targetAttribute' => ['doctor_id' => 'id']],
            [['client_id'], 'exist', 'skipOnError' => true, 'targetClass' => User::class,
'targetAttribute' => ['client_id' => 'id']],
        ];
    }

    public function attributeLabels(): array
    {
        return [
            'id' => 'ID',
            'doctor_id' => 'Лікар',
            'client_id' => 'Клієнт',

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				88
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'rating' => 'Оцінка',
        'comment' => 'Коментар',
        'date' => 'Дата',
    ];
}

public static function createReview($data)
{
    $review = new Review();
    $review->doctor_id = $data['doctor_id'];
    $review->client_id = $data['client_id'];
    $review->rating = $data['rating'];
    $review->comment = $data['comment'];
    if (!$review->save()) {
        Yii::error($review->getErrors(), 'review-create');
        return false;
    } else {
        return $review;
    }
}

public static function deleteReview($id)
{
    $review = self::getReviewById($id);
    if (!$review) {
        return null;
    }
    if (!$review->delete()) {
        Yii::error('Failed to delete review' . json_encode($review->getErrors()),
'session-delete');
        return null;
    } else {
        Yii::$app->session->setFlash('success', 'Review deleted successfully.');
```

```

        return $review;
    }
}

public static function getReviewsByDoctorId($doctorId, $order = 'SORT_DESC')
{
    return self::find()
        ->where(['doctor_id' => $doctorId])
        ->orderBy(['rating' => $order])
        ->all();
}

public static function getReviewById($id)
{
    return self::findOne(['id' => $id]);
}

public static function getAverageRating($doctorId): ?float
{
    return Review::find()
        ->where(['doctor_id' => $doctorId])
        ->average('rating');
}

public function getDoctor()
{
    return $this->hasOne(User::class, ['id' => 'doctor_id']);
}

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				89
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public function getClient()
    {
        return $this->hasOne(User::class, ['id' => 'client_id']);
    }
}

```

Лістинг Article.php:

```

<?php

namespace app\models;

use Yii;
use yii\db\ActiveRecord;
use yii\helpers\ArrayHelper;

/**
 * This is the model class for table "article".
 *
 * @property int $id
 * @property int $doctor_id
 * @property string $created_at
 * @property string $updated_at
 * @property string $content
 * @property string $status
 *
 * @property User $doctor
 */
class Article extends ActiveRecord
{
    const STATUS_REVIEWING = 'reviewing';
    const STATUS_APPROVED = 'approved';
    const STATUS_REJECTED = 'rejected';

    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'article';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['title', 'doctor_id', 'content'], 'required'],
            [['doctor_id'], 'integer'],
            [['created_at', 'updated_at'], 'safe'],
            [['title', 'content'], 'string'],
            [['status', 'in', 'range' => array_keys(self::statusList())],
            [['status'], 'default', 'value' => self::STATUS_REVIEWING],
            [['doctor_id'], 'exist', 'skipOnError' => true, 'targetClass' => User::class,
            'targetAttribute' => ['doctor_id' => 'id']],

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмоторя О.В.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    ];
}

/**
 * {@inheritdoc}
 */
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'doctor_id' => 'Лікар',
        'created_at' => 'Дата створення',
        'updated_at' => 'Дата оновлення',
        'title' => 'Заголовок',
        'content' => 'Контент',
        'status' => 'Статус',
    ];
}

public function getDoctor()
{
    return $this->hasOne(User::class, ['id' => 'doctor_id']);
}

public static function statusList(): array
{
    return [
        self::STATUS_REVIEWING => 'На перевірці',
        self::STATUS_APPROVED => 'Схвалено',
        self::STATUS_REJECTED => 'Відхилено',
    ];
}

public function getStatusName(): string
{
    return self::statusList()[$this->status];
}

public static function create(array $data): ?self
{
    $article = new self();
    if (!$article->validate()) {
        Yii::error($article->getErrors(), 'article-create');
        return null;
    }
    $article->doctor_id = $data['doctor_id'];
    $article->title = $data['title'];
    $article->content = $data['content'];
    return $article->save ? $article : null;
}

public function updateArticle(array $data): bool
{
    if ($this->load($data, '') && $this->save()) {
        return true;
    }
    Yii::error($this->getErrors(), 'article-update');
    return false;
}

```

```

public function updateStatus($status): bool
{
    $this->status = $status;
    if( $this->save()) {
        return true;
    }
    Yii::error($this->getErrors(), 'article-update-status');
    return false;
}

public function deleteArticle(): bool
{
    return $this->delete() !== false;
}

public static function getById($id): ?self
{
    return self::findOne($id);
}

public static function getAllByDoctor($doctorId)
{
    return self::find()->where(['doctor_id' => $doctorId])->orderBy(['updated_at' =>
SORT_DESC]);
}

public static function getAllApproved()
{
    return self::find()->where(['status' => self::STATUS_APPROVED])-
>orderBy(['updated_at' => SORT_DESC]);
}

public static function getAllRejected()
{
    return self::find()->where(['status' => self::STATUS_REJECTED])-
>orderBy(['updated_at' => SORT_DESC]);
}

public static function getAllReviewing()
{
    return self::find()->where(['status' => self::STATUS_REVIEWING])-
>orderBy(['updated_at' => SORT_DESC]);
}

public static function getAllByStatus($status)
{
    return self::find()->where(['status' => $status])->orderBy(['updated_at' =>
SORT_DESC]);
}
}

```

		Піщук С.О.			Житомирська політехніка.25.121.20.000 - ПЗ	Арк.
		Чижмотря О.В.				92
Змн.	Арк.	№ докум.	Підпис	Дата		