

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

БЕГЛОВА АЛЬБІНА ВАДИМІВНА

Допускається до захисту:

В.о. завідувача кафедри
інформаційних технологій,
канд. техн. наук, доцент

Оксана ЗЕЛІНСЬКА

«_____» _____ 2024р.

**РОЗРОБКА ТА ОПТИМІЗАЦІЯ СЕРВЕРНОЇ АРХІТЕКТУРИ ДЛЯ
ВЕБПЛАТФОРМИ ПСИХОТЕРАПЕВТИЧНИХ ПОСЛУГ**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Павло РИМАР, старший викладач
кафедри інформаційних технологій

Оцінка: _____ / _____ / _____
(бали за шкалою ЄКТС / за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця – 2024

АНОТАЦІЯ

Бєглова А.В Розробка та оптимізація серверної архітектури для вебплатформи психотерапевтичних послуг. Спеціальність 122 «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

Основною метою виконання кваліфікаційної роботи є проектування та реалізація ефективної серверної архітектури з її подальшою оптимізацією.

У вступі розглядається проблематика знецінення ментального здоров'я, важливість цифровізації медицини, особливо психотерапії. У першому розділі наведено план початкової реалізації та існуючі аналоги продукту. У другому розділі розглянуто інструменти для розробки та оптимізації. Третій розділ присвячений програмній реалізації.

Ключові слова: Backend, Webservice, PHP, Database

ABSTRACT

A. Behlova Development and optimization of server architecture for the web platform of psychotherapeutic services. Specialty 122 "Computer Science". Vasyl Stus Donetsk National University, Vinnytsia, 2024.

The main purpose of the qualification work is the design and implementation of an effective server architecture with its further optimization.

The introduction examines the issue of devaluation of mental health, the importance of digitalization of medicine, especially psychotherapy. In the first section, the initial implementation plan and existing analogues of the product are given. The second section discusses tools for development and optimization. The third section is devoted to software implementation.

Keywords: Backend, Webservice, PHP, Database

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ.....	4
ВСТУП	5
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ З РОЗРОБКИ КОМПОНЕНТІВ ВЕБ- СЕРВІСУ ТА ЇХ ОПТИМІЗАЦІЇ.....	7
1.1 Початкова реалізація.....	7
1.2 Аналоги продукту	7
Висновок до розділу 1	17
РОЗДІЛ 2. ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ РОЗРОБКИ ТА ОПТИМІЗАЦІЇ	18
2.1 Розробка продукту	18
2.2 Оптимізація роботи продукту.....	20
Висновок до розділу 2	28
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	29
3.1 Опис бази даних	29
3.2 Програмна реалізація.....	34
Висновок до розділу 3	38
ВИСНОВКИ.....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

Скорочення	Повна назва	Пояснення
API [1]	Application Programming Interface	Програмний інтерфейс застосунку/веб-продукту
SSL/TLS [2]	Secure Sockets Layer та Transport Layer Security	Сучасні протоколи криптографічного захисту
WS [3]	Webservice	Сервіс, що допомагає різним програмам взаємодіяти
CSS [4]	Cascading Style Sheets	мова стилізації
CDN [5]	Content Delivery Network	новітня мережа доставки контенту
JS [6]	JavaScript	мова програмування для інтерактиву всередині браузеру
HTML [7]	HyperText Markup Language	основна мова розмітки
CSS Sprites [8]	Cascading Style Sheets Image Sprites	Метод оптимізації рендеру зображень на веб-сторінці
TS [9]	TypeScript	мова програмування для бекенд-складової
-	Angular [10]	Фреймворк, написаний на TS
PHP [11]	Hypertext reprocessor	скриптова мова програмування
SQL [12]	Structured Query Language	мова програмування для роботи з БД
XML [13]	eXtensible Markup Language	мова розмітки для зберігання та передачі даних
JSON [14]	JavaScript Object Notation	форматування даних для обміну ними

ВСТУП

Сучасний світ характеризується стрімким розвитком технологій, що безпосередньо впливає на всі сфери людської діяльності, включаючи медицину та психотерапію. Різні світові зворушення, такі як пандемія COVID-19 та повномасштабне вторгнення росії в Україну спричинили значне зростання попиту на психотерапевтичні онлайн-послуги, оскільки багато людей наразі відчувають значний рівень стресу та тривожності. Однак, зростаюча проблема в онлайн-психотерапії очевидно ставить перед технологічною сферою нові виклики, зокрема щодо створення ефективних та захищених серверних архітектур, здатних забезпечувати надійний доступ до терапії через Інтернет тим, хто цього потребує.

Завданням дослідження є внесок у «цифровізацію» медицини, особливо психотерапії – сучасний елемент підвищення доступності та ефективності охорони здоров'я в умовах глобалізації та технологічного прогресу. Переведення терапії в мережу Інтернет дозволяє пацієнтам отримувати допомогу безпосередньо в своїх домівках, забезпечуючи зручність і приватність, відчиняючи двері в терапію самим вразливим наразі верствам населення – жертвам війни. Новітні інструменти, такі як мобільні додатки, відеоконсультації та онлайн-сеанси терапії поєднують в собі простоту використання та відчуття безпеки під час візиту, допомагають пацієнтові побороти стигму та бар'єри, пов'язані зі скептичним відношенням до ментального здоров'я. Цифровізація сприяє кращому управлінню медичною інформацією та даними пацієнтів, веденням різноманітних звітностей, залишивши позаду «медичні картки» та застарілі паперові архіви. Аналіз великих обсягів даних саме в цифровому вигляді сприяє більш індивідуалізованому підходу в лікуванні кожного пацієнта. В умовах індустріалізованого підходу до медицини, професіонали у сфері психотерапії можуть зарекомендувати себе у сучасному медичному просторі, мають змогу навчатися та обмінюватися досвідом на

міжнародному рівні, це в свою чергу сприяє розвитку нових методів лікування ментальних недуг та стратегій взаємодії з проблемними пацієнтами.

У рамках бакалаврської роботи мета полягає в наступному:

1. Поглиблене вивчення сучасних технологій створення та оптимізації вебсервісів.
2. Огляд існуючих додатків та веб-продуктів з тематикою «психотерапевтичні послуги».
3. Написання серверної частини веб-ресурсу для надання послуг терапії.

Задачами дослідження є:

- Огляд існуючих сайтів з надання психотерапевтичних послуг (з можливістю спеціалісту зареєструватися як надавач конкретних послуг), виявлення їх переваг та недоліків, проблематика подібних порталів.
- Вивчення та використання спеціалізованого програмного забезпечення та фреймворків для розробки.
- Розробка серверної частини, налагодження процесу взаємодії сайту з базою даних, оптимізація та перевірка роботи API.

Об'єкт дослідження: серверна частина веб-продукту з надання психотерапевтичних послуг.

Предмет дослідження: процес розробки та оптимізації серверної частини сайту.

Структура роботи: кваліфікаційна робота складається зі вступу, трьох розділів, висновків та списку використаних джерел. Робота містить 40 сторінок, 22 рисунків, 7 таблиць та список літератури з 12 джерел.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ З РОЗРОБКИ КОМПОНЕНТІВ ВЕБ-СЕРВІСУ ТА ЇХ ОПТИМІЗАЦІЇ

1.1 Початкова реалізація

Необхідно розробити API [1] складову проекту з вже існуючою фронтенд частиною. Веб-портал виконуватиме наступні функції:

1. Має сучасний інтуїтивно зрозумілий дизайн.
2. Є крос-браузерним та оптимізованим для слабких пристроїв.
3. Адаптований до користування на телефонах, підтримує різні оперативні системи (Android, IOS, Windows Mobile)
4. Сайт імплементує SSL/TLS [2] для шифрування з'єднань.
5. Можливість реєстрації з подальшим заходом на сайт використовуючи свої облікові дані що підв'язуються до токenu – унікального коду, що складається з випадкових комбінацій цифр та літер латиниці.
6. Система запису на прийом, з вікном для вибору спеціаліста та тривалості сеансу, з можливістю заповнення анкети, яка включає в себе облікові дані, діагноз за його наявності, вік та стать. Клієнт одразу може дізнатися вартість майбутньої послуги.
7. Модуль для зворотнього зв'язку.
8. Різні варіанти сеансів – запис для одного, терапія для пар, терапія батько/дитина.

1.2 Аналоги продукту

Перші веб-продукти з надання психологічних та психотерапевтичних послуг почали з'являтися на початку нульових років, в Україні це відбулося дещо пізніше – після 2010 року.

Серед непоганих іноземних аналогів можна виділити 3 найвідоміших продукти:

- <https://www.talkspace.com>
- <https://www.talkiatry.com>

- <https://www.betterhelp.com>

Розглянемо кожну платформу:

1. Talkspace

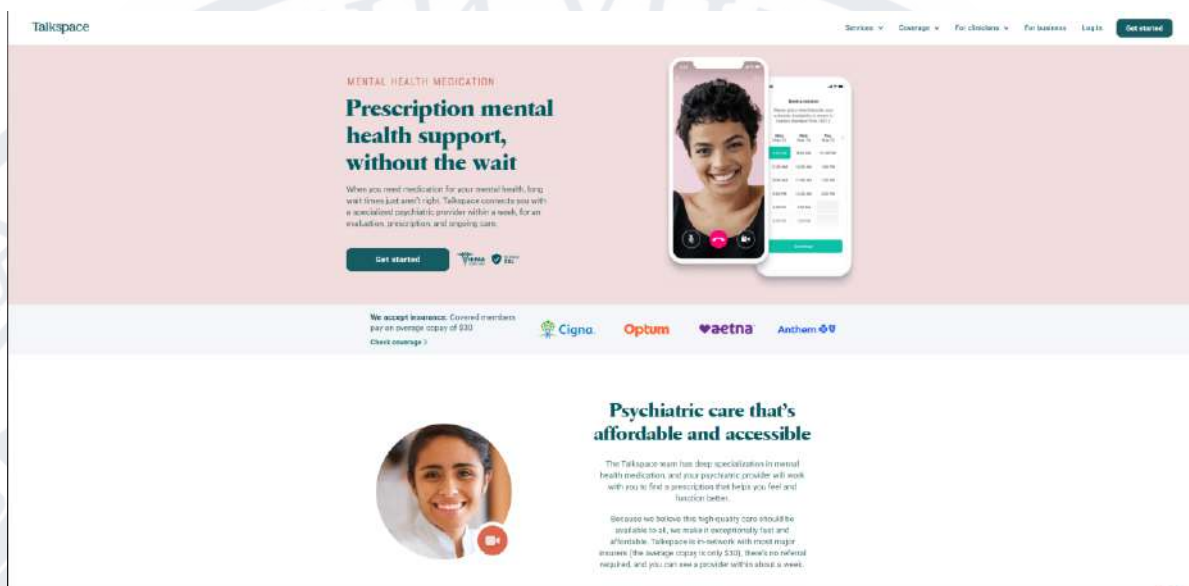


Рисунок 1.1 Головна сторінка сайту «Talkspace»

Talkspace – один з найбільших в США онлайн-ресурсів з надання психотерапевтичних послуг. Основною його перевагою серед аналогів є інтуїтивно зрозумілий дизайн, що поєднується з великим різноманіттям послуг – від підліткового психотерапевта до психіатрії тяжких ментальних хвороб. Сайт пропонує пройти невеличке персоналізоване опитування, яке в результаті підбирає з великої бази спеціалістів саме необхідного клієнтові. Основними недоліками Talkspace є велика ціна сеансу зі спеціалістом та орієнтованість лише на Південну Америку.

Переглянувши код сторінки можна неозброєним оком помітити, що на сайті використовується метод оптимізації рендеру зображень CSS Sprites [8]. Всі картинки та іконки які є на сторінці, початково об'єднані в один великий лист, якщо в штучних умовах зменшити швидкість інтернету, можна помітити як картинки які з'являються в полі зору на екрані починають вивантажуватись з кешу на сайт.

Бекенд-частина даного продукту написана переважно на TypeScript, з використанням бібліотек jQuery та Slick.

2. Talkiatry

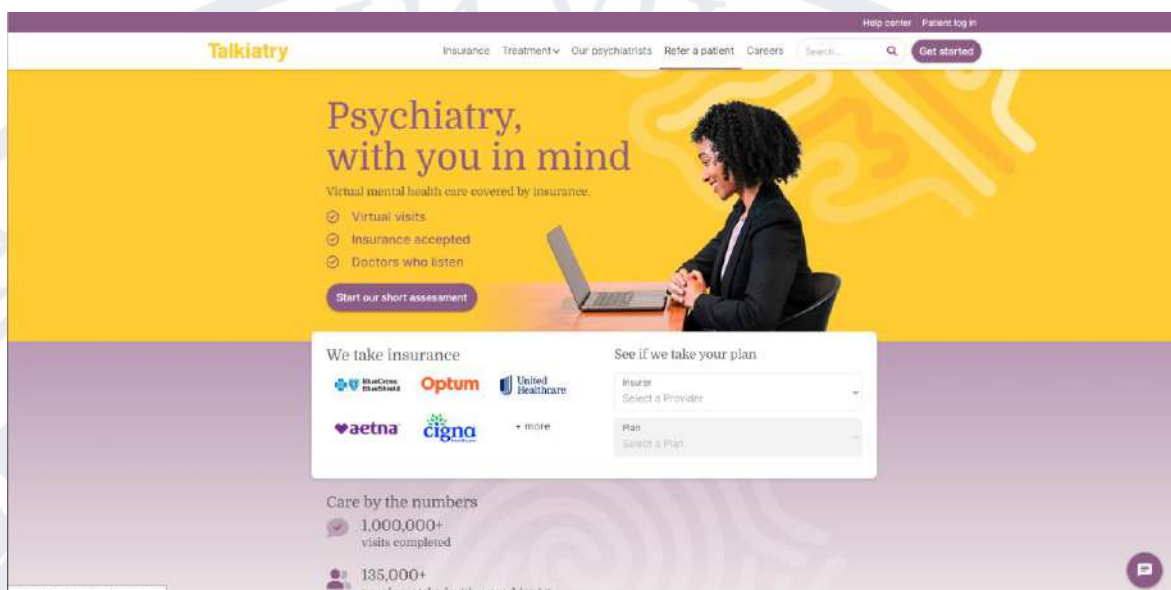


Рисунок 1.2 Головна сторінка сайту «Talkiatry»

Talkiatry є ще одним Американським ресурсом онлайн психології та психіатрії. Він більш перевірений часом ніж попередній, але через це має суттєвий недолік – незрозумілий та негармонійний дизайн, його кольорова палітра збиває з пантелику початкового відвідувача, а велика кількість іконок страховиків з якими вони співпрацюють збиває з пантелику. Перевагами цього продукту є доступні ціни та більша база спеціалістів, ніж в Talkiatry. Процедура реєстрації вимагає лише двох кліків.

Так як цей ресурс орієнтований не лише на Америку, а й охоплює Європу та частину східної Азії, задля оптимізації онлайн сеансів в режимі реального часу зі спеціалістом програмісти сайту скористалися CDN [5]. Сервери Talkiatry розміщені в різних точках світу, близько біля регіонів з піковою відвідуваністю сайту, завдяки цьому сайт швидко завантажується, не підвисає, у відеочатах з лікарем зображення не здригається та є досить чітким.

Бекенд складова написана на TypeScript [9] з використанням Angular [10].

3. Betterhelp

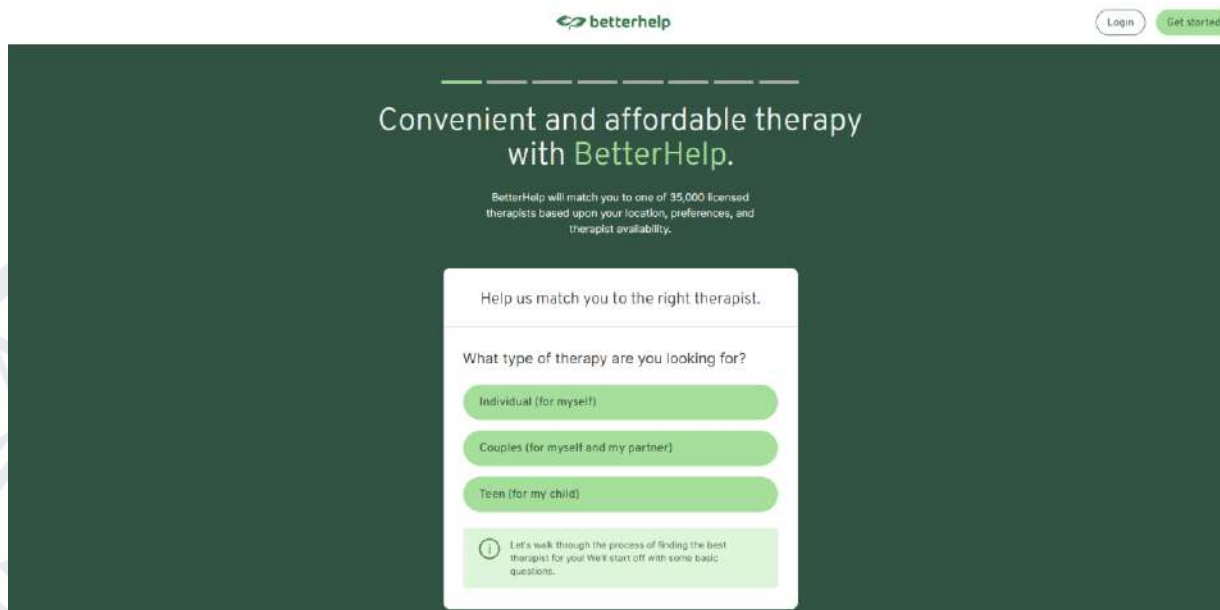


Рисунок 1.3 Головна сторінка сайту «Betterhelp»

Betterhelp – орієнтований переважно на аудиторію Європи сайт з отримання психотерапевтичних послуг. Пропонує багату вибірку спеціалістів зі всього світу, за доволі низьку платню, яка може становити нуль за умови що освітній заклад/місце роботи потенційного пацієнта співпрацює з Betterhelp. Його найбільшою перевагою є неймовірна оптимізація ресурсу, він миттєво завантажується, перехід між сторінками порталу відбувається блискавично. Процедура реєстрації, яка «під капотом» являє собою заведення великої кількості нових даних в базу даних – відбувається за лічені секунди. При створенні порталу використано багато різноманітних прийомів з оптимізації, можна помітити наступні:

- **CDN [5]**
- **CSS Sprites [8]**
- **Нормалізація БД** – оскільки сайти на кшталт вище зазначених зберігають в своїх базах даних багато прізвищ, імен, анамнезів, необхідно забезпечити цілісність та оптимізацію усіх даних. В такому випадку нормалізація забезпечує високий рівень цілісності даних, оскільки зменшує ймовірність помилок вводу та забезпечує правильне пов'язування даних у таблицях.

Отримана в результаті нормалізована БД спрощує управління даними, тому що зміни у одному місці бази даних автоматично відображаються в інших частинах. Це дуже важливо для онлайн-психотерапії, де дані про пацієнтів часто оновлюються і мають бути точними.

- При користуванні будь-яким пошуковим полем, а також функцією реєстрації на сайт, можна помітити що на сайті використовується **індексація полів**.
- **Кешування.** Сайт кешує збережені результати запитів, часто використовуваних даних або ресурсів задля покращення швидкості відповіді сервера. При цьому в кеші зберігається інформація про терапевтів, відгуки про сеанс терапії та різноманітні статті.
- **Використання мікросервісної архітектури:** Програмісти сайту делегували різні функціональності сайту окремим одиницям, що називаються мікросервіси – невеличкі програми, незалежні одна від одної, які разом складають цілі проекти та функціональності. Використання мікросервісів значно покращує масштабованість та легкість управління. Кожен мікросервіс виконує окрему задачу, на кшталт управління сесіями, автентифікації або записи на прийом.
- **Асинхронне програмування:** Використання асинхронних підходів у роботі з мережевими запитами або довгими процесами допомагає уникнути блокування сервера для звичайних користувачів і покращує його репутацію в пошукових мережах, Google, Yahoo тощо.

Україна також розвивається в напрямку цифровізації своєї медицини, особливо психотерапії та психології. До найвідоміших вітчизняних психотерапевтичних ресурсів можна віднести наступні:

- <https://www.rozmova.me>
- <https://e-likari.com.ua/doctors/psixoterapevt/>
- <https://pleso.me/ua>

Розглянемо кожен сайт:

1. Rozmova

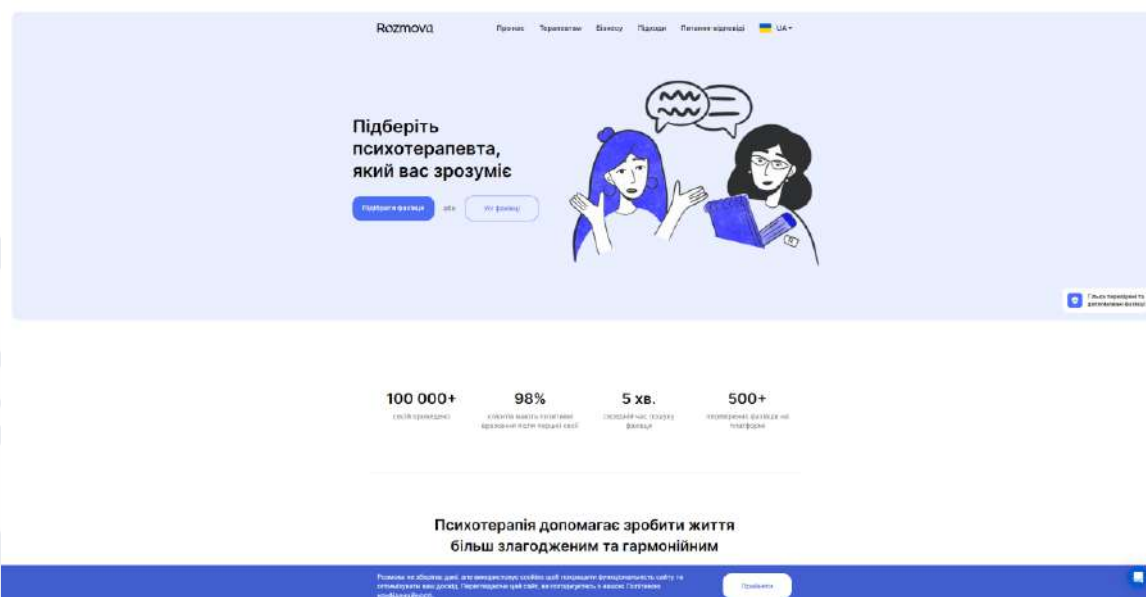


Рисунок 1.4 Головна сторінка сайту «Rozmova»

Загальний вигляд сайту створює дуже приємне враження завдяки своїй світлій колірній гамі, яка використовує блакитні та білі відтінки. Це робить сайт візуально привабливим і створює спокійне відчуття, що дуже важливо для платформи, яка надає психологічну допомогу. Ілюстрації на головній сторінці виконані у простому стилі, що допомагає зосередитися на основному контенті без відволікання уваги.

Навігація на сайті є інтуїтивно зрозумілою. Верхнє меню чітко видиме і містить основні розділи, що полегшує переміщення по сайту. Кнопки, зокрема "Підібрати фахівця", виділяються синім кольором, що привертає увагу користувачів і спонукає до дії. Щодо контенту, то статистичні дані, такі як кількість проведених сесій та рівень задоволеності клієнтів, розміщені у вигляді інформаційних блоків. Це добре структурує інформацію і робить її легкою для сприйняття. Крім того, наявність чітких закликів до дії мотивує користувачів скористатися послугами сайту.

Сайт використовує сучасні технології, такі як TypeScript, Angular та JQuery. Використання TypeScript знижує кількість помилок і підвищує надійність коду завдяки статичній типізації. Angular забезпечує модульність та

структурованість додатку, що полегшує його підтримку та розвиток. JQuery може бути використаний для специфічних завдань, хоча сучасні фреймворки, такі як Angular, часто можуть обійтися без нього.

Функціональність сайту значно підвищується завдяки використанню Angular, що дозволяє створювати динамічні інтерфейси, які швидко реагують на дії користувача. Це забезпечує високий рівень взаємодії з користувачем та покращує загальний користувацький досвід. Використання сучасних фреймворків та мов програмування також сприяє швидкому завантаженню сторінок, що є важливим для утримання уваги користувачів.

Варто також зазначити кілька загальних рекомендацій для покращення сайту. По-перше, варто переконатися, що дизайн є адаптивним і зручним для використання на мобільних пристроях, оскільки багато користувачів будуть звертатися до сайту саме з мобільних телефонів. По-друге, додавання більше інтерактивних елементів може значно покращити користувацький досвід. Що стосується бекенду, важливо переконатися, що він оптимізований для швидкого обслуговування великої кількості запитів. Особливу увагу варто приділити безпеці, враховуючи чутливий характер даних, з якими працює сайт.

Таким чином, загальний вигляд сайту створює приємне враження, використання сучасних технологій на бекенді забезпечує його надійність та функціональність. В поєднанні з доступністю послуг та різноманітністю спеціалістів всі вищезазначені пункти роблять сайт «Rozмова» найкращим існуючим порталом з надання психотерапевтичних послуг.

2. E-Likari

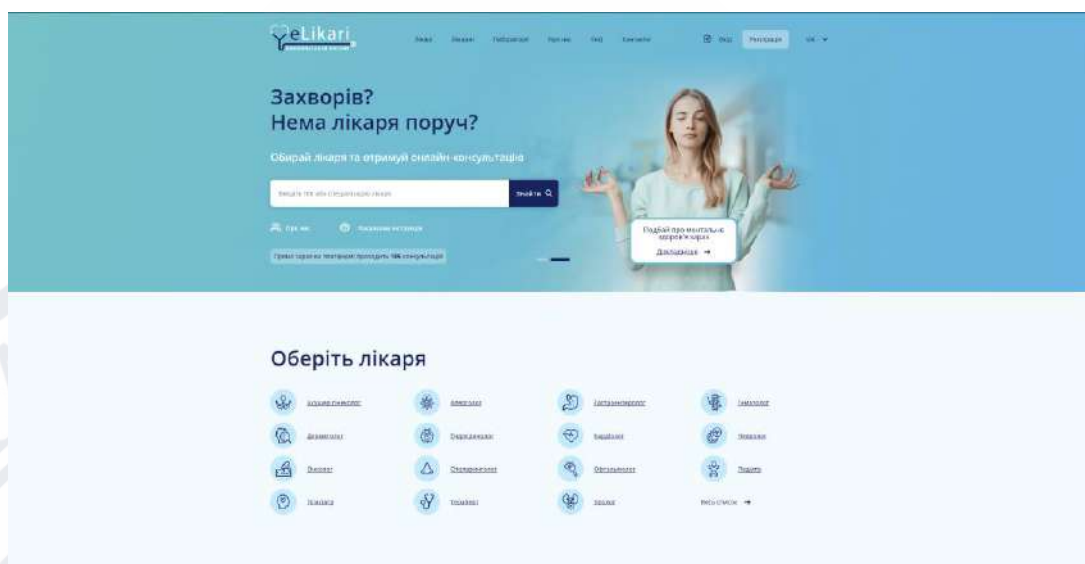


Рисунок 1.5 Головна сторінка сайту «E-Likari»

E-Likari надає послуги загальної онлайн-медицини по всій Україні, не обмежуючись лише психологами та психотерапевтами.

Дизайн сайту для надання онлайн-медичних послуг виглядає професійно та зручним для користувачів. Головна сторінка має чистий і сучасний вигляд з приємною кольоровою гамою, яка сприяє створенню довіри до медичних послуг. Використання білого і світло-блакитного кольорів, а також зображення лікарів у білих халатах підсилює відчуття професіоналізму та надійності.

Верхня частина сторінки містить зрозумілий і зручний для користувача інтерфейс з пошуковим полем, де можна ввести прізвище або спеціалізацію лікаря для швидкого пошуку. Кнопки навігації чітко виділені і легко доступні, що полегшує переміщення по сайту. Є можливість швидкої реєстрації та входу в систему, що спрощує процес для нових та постійних користувачів.

Сайт пропонує відразу кілька основних переваг: різноманітність лікарів з різними стажами роботи та різною вартістю сеансу, консультація без черг та можливість почати безкоштовний чат з бажаним спеціалістом, що є важливими для користувачів, які шукають негайну медичну допомогу. Розділ "Обери лікаря" розташований нижче, надає чіткий та зручний доступ до різних медичних спеціальностей за допомогою іконок та назв.

Інтерфейс побудований на сучасних технологіях, таких як Node.js, Nuxt.js і Vue.js, що забезпечує високу продуктивність, гнучкість та швидку реакцію сайту на дії користувачів. Це також гарантує можливість масштабування та подальшого розвитку платформи відповідно до потреб користувачів та нових технологічних тенденцій.

Загалом, дизайн сайту добре продуманий з точки зору користувача, забезпечуючи легкість навігації, доступність важливої інформації та професійний вигляд, що є критично важливим для онлайн-медичних послуг. Використані програмістами сучасні технології для написання бекенд-частини ресурсу забезпечили його великою кількістю задоволених користувачів по всій Україні.

3. Pleso

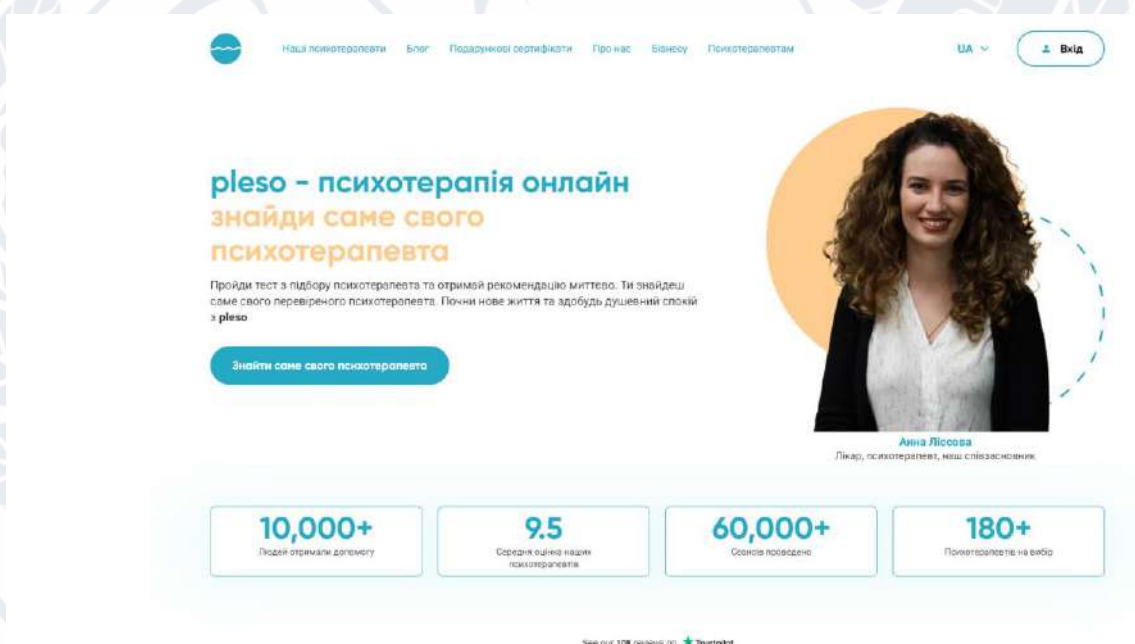


Рисунок 1.6 Головна сторінка сайту «Pleso»

Pleso – Український ресурс для надання послуг психотерапії. Надає можливість пройти коротке персоналізоване опитування, в процесі якого потенційний користувач повинен обрати вид послуги яку бажає отримати: парну терапію або сеанс для одного. Серед недоліків можна відмітити низьку

кваліфікацію спеціалістів по середній вибірці, високу вартість консультації відносно аналогів в Україні. Відсутність терапії виду батько-дитина та довгу процедуру реєстрації. Попри те, Pleso використовує комбінацію сучасних технологій для своєї фронтенд та бекенд частини.

Фронтенд реалізований за допомогою React, популярної бібліотеки для побудови користувацьких інтерфейсів, дозволяє створювати динамічні та інтерактивні компоненти, що забезпечує плавний користувацький досвід в поєднанні з сучасним виглядом сторінок сайту. Використання Emotion як бібліотеки для стилізації надає можливість писати CSS безпосередньо в JavaScript, що спрощує процес створення і підтримки стилів. Це рішення забезпечує високу продуктивність та легкість у підтримці і розвитку інтерфейсу, дозволяючи швидко реагувати на зміни потреб користувачів.

Бекенд частина реалізована на PHP, популярній мові програмування для веб-розробки завдяки своїй простоті, широкій підтримці та великій кількості доступних фреймворків і бібліотек. Використання PHP дозволяє швидко обробляти запити користувачів і ефективно управляти базами даних. Інтеграція з фронтендом через RESTful API або GraphQL забезпечує швидкий та надійний обмін даними між клієнтською та серверною частинами.

Порівнюючи цей портал з іншими аналогами в Україні, можна відзначити кілька ключових моментів. Багато українських платформ, що надають подібні послуги, використовують різні технології для побудови своїх сайтів. Наприклад, деякі платформи можуть використовувати Angular або Vue.js для фронтенд частини, що також забезпечує високий рівень інтерактивності, але має інші підходи до організації коду та управління станом. Використання таких технологій як Vue.js може забезпечувати простішу криву навчання для розробників, тоді як Angular пропонує більш структуроване і масштабоване рішення для великих проектів.

Для бекенду, окрім PHP, часто використовуються Node.js та Python. Node.js забезпечує асинхронну обробку запитів, тим самим підвищуючи продуктивність і швидкість відповіді серверу, особливо при великій кількості одночасних

запитів. Python, завдяки своїм потужним фреймворкам, таким як Django чи Flask, забезпечує високий рівень безпеки і гнучкості в розробці веб-додатків.

Дизайн і користувацький досвід також є важливими аспектами. Сайт, що розглядається, має сучасний і привітний інтерфейс, що відображає тенденції у веб-дизайні, такі як мінімалізм, використання великих шрифтів і яскравих кольорів. Інші українські платформи можуть мати більш традиційний дизайн, що може здаватися менш привабливим для молодшої аудиторії, але може бути більш знайомим і комфортним для старших користувачів.

Загалом, **Pleso.me** демонструє високу якість технічної реалізації та привабливий дизайн, що робить його конкурентоспроможним на ринку онлайн психотерапевтичних послуг в Україні. Він використовує сучасні технології, що забезпечують швидкість, надійність і зручність користувача, що є критично важливим для залучення та утримання клієнтів.

Висновок до розділу 1

В першому розділі було сформульовано постановку задачі з початковою реалізацією серверної архітектури для порталу надання психотерапевтичних послуг. Згодом було розглянуто декілька найвідоміших світових (США, центральна Європа) та вітчизняних аналогів. Як результат огляду аналогів чітко відслідковується проблематика недостатньої оптимізованості бекенд частини у вітчизняних психотерапевтичних сайтах, що вкотре підтверджує актуальність та потенційний попит до створення нового ресурсу з урахуванням помилок існуючих сервісів.

РОЗДІЛ 2

ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ РОЗРОБКИ ТА ОПТИМІЗАЦІЇ

2.1 Розробка продукту

Сучасний світ пропонує безліч різноманітних мов програмування та фреймворків для розробки бекенд частини сайту. Задля оптимізації ресурсу шляхом використання мікросервісної архітектури та експерименту з визначення швидкодії окремих обраних утиліт, для кожного модулю на сайті обрано різні утиліти, такі як:

PHP [11]



Рисунок 2.1 Логотип PHP

PHP є мовою сценаріїв загального призначення, яка широко використовується для розробки вебдодатків та динамічних вебсайтів. У бекенд розробці PHP відіграє важливу роль, оскільки дозволяє створювати серверні частини додатків, обробляти запити від користувачів, взаємодіяти з базами даних і генерувати HTML-сторінки, що відображаються у браузері. Має кілька переваг: він простий у вивченні та використанні, його основи легко засвоїти будучи зовсім новачком в програмуванні. Також PHP має безліч готових бібліотек та фреймворків, які значно спрощують розробку складних додатків.

MongoDB



Рисунок 2.2 Логотип mongoDB

MongoDB – це система управління базами даних, яка належить до категорії NoSQL і використовує документи у форматі BSON (бінарна версія JSON[14]) для зберігання даних. Основною перевагою MongoDB є гнучка структура даних. Дані зберігаються у вигляді текстових документів. При цьому система забезпечує високу продуктивність завдяки використанню індексів, кешування та інших механізмів оптимізації. MongoDB підтримує горизонтальне масштабування через шардінг, дозволяючи при цьому розподіляти дані на кілька серверів, а також реплікацію для забезпечення високої доступності та відмовостійкості.

phpMyAdmin



Рисунок 2.3 Логотип phpMyAdmin

phpMyAdmin є безкоштовним інструментом із відкритим кодом, написаний на PHP, який надає веб-інтерфейс для керування базами даних . За

допомогою нього можна виконувати різноманітні операції з базою даних через веб-браузер, роблячи керування базою даних доступним і зручним без вимагання особливих пізнань в SQL [12].

Цей інструмент дозволяє створювати, редагувати та видаляти бази даних і таблиці, а також виконувати SQL-запити безпосередньо через інтерфейс. phpMyAdmin підтримує імпорт та експорт даних у різних форматах, таких як SQL та XML [13], що робить його зручним для резервного копіювання та міграції даних. Крім того, він дозволяє вам керувати правами доступу користувачів до баз даних, створювати та редагувати індекси для забезпечення продуктивності запитів і переглядати статистику бази даних, таку як кількість рядків у таблицях і розмір бази даних.

Розробники та адміністратори баз даних часто використовують phpMyAdmin, щоб легко та швидко налаштовувати нові бази даних, зручно редагувати структуру бази даних і дані в таблицях, виконувати резервне копіювання та відновлення баз даних, а також керувати правами доступу та безпекою бази даних. Завдяки веб-інтерфейсу phpMyAdmin можна використовувати на будь-якому комп'ютері з доступом до Інтернету, що робить його дуже зручним інструментом для віддаленого керування базами даних.

2.2 Оптимізація роботи продукту

Для оптимізації роботи сайту будуть застосовані наступні засоби:

1. Оптимізація Баз даних, для покращення швидкості пошуку даних (на формі підбору спеціаліста, формі авторизації на сайт тощо) будуть застосовані індекси з регулярною нормалізацією бази даних для усунення надмірності даних.



Рисунок 2.4 Приклад роботи нормалізації БД

- Індексування ключових полів в БД. Критично важлива операція для оптимізації запитів, що надходять до бази даних. Індекс у базі даних схожий на індекс у книзі: він допомагає системі швидко знаходити дані без необхідності перевіряти кожен рядок таблиці при кожному запиті. Індeksi значно прискорюють пошук даних, особливо при великих обсягах даних. Важливо прискіпливо вибирати поля для індексування. Ними повинні бути лише поля, значення яких часто використовуються в умовах пошуку або сортування, оскільки кожен індекс вимагає додаткового простору на накопичувачі та може уповільнити записи в базу даних.
- Для часто використовуваних запитів, що надсилаються на сервер застосовуватиметься кешування, що зберігатиме результати цих самих запитів. В залежності від ситуації застосовуватиметься як локальне кешування (на сервері), так і кешування на стороні клієнта.
- Для оптимізації роботи сайту на слабких пристроях (старі моделі мобільних телефонів, комп'ютери на застарілих версіях Windows тощо)

використовуватиметься CSS [4] Sprites – перевірена часом техніка оптимізації веб-сторінок, яка дозволяє зменшити кількість HTTP-запитів, які браузер надсилає під час завантаження сторінки. Замість того, щоб завантажувати багато окремих зображень, CSS Sprites об'єднує ці зображення в одне велике зображення. Потім за потреби можна вказати, яку частину цього великого зображення відобразити в певному місці на сторінці. Особливо корисною ця технологія себе демонструє коли потрібно прогрузити на сторінці кнопки або іконки, які застосовуються всюди.



Рисунок 2.5 Приклад CSS Sprites [8]

5. Компоненти сайту можна завантажувати асинхронно, це необхідно для того щоб веб-сторінки могли продовжувати завантажуватись на стороні клієнта навіть від час завантаження скриптів. З цією ж метою імплементується так званий «Lazy Loading» - техніка оптимізації веб-сайтів та додатків, яка полягає у відкладеному завантаженні ресурсів або об'єктів до того моменту, коли вони дійсно потрібні. За допомогою такого способу завантаження значно покращується час рендеру сторінки, ефективність використання порталу, зменшується навантаження на сервер.

6. Використання CDN [5] - системи розподілених серверів, які допомагають швидко доставляти інтернет-контент (веб-сторінки, відео, зображення, стилі та JavaScript-файли) користувачам на основі їхнього географічного розташування. Сервери CDN розміщені по всьому світу, і коли користувач запитує якийсь контент, він отримується з найближчого до нього сервера. З розвитком проекту та перспективою виходу на міжнародний хостинг з'явиться потреба в використанні CDN. Основні переваги CDN включають зменшення затримок завдяки розміщенню серверів ближче до користувачів, підвищення надійності шляхом розподілу трафіку між багатьма серверами, що забезпечує стабільність роботи навіть під час пікових навантажень чи DDoS-атак. Використання CDN дозволяє швидше завантажувати веб-сторінки, що позитивно впливає на SEO та залученість користувачів. Також CDN оптимізує трафік, автоматично вибираючи найоптимальніший маршрут для передачі даних, що зменшує затримки та втрати пакетів. Також CDN вміє обробляти як статичний, так і динамічний контент. Статичний контент включає зображення, CSS, JavaScript, відео, аудіо та інші типи даних, які рідко змінюються. Динамічний контент генерується в режимі реального часу і включає користувацькі профілі, результати пошуку, персоналізовані рекомендації. Хоча динамічний контент важче кешувати, сучасні CDN оптимізують його доставку за рахунок маршрутизації та зменшення затримок.

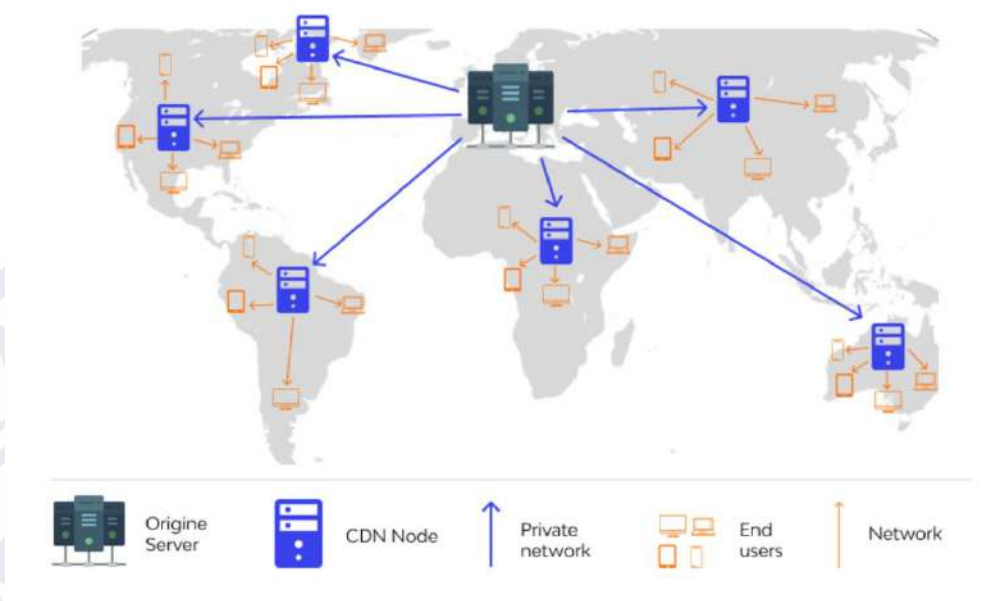


Рисунок 2.6 Ілюстрація роботи CDN [5]

7. Перед передачею на сторону клієнта HTML, CSS та JS файлів вони будуть стиснені за допомогою GZIP.

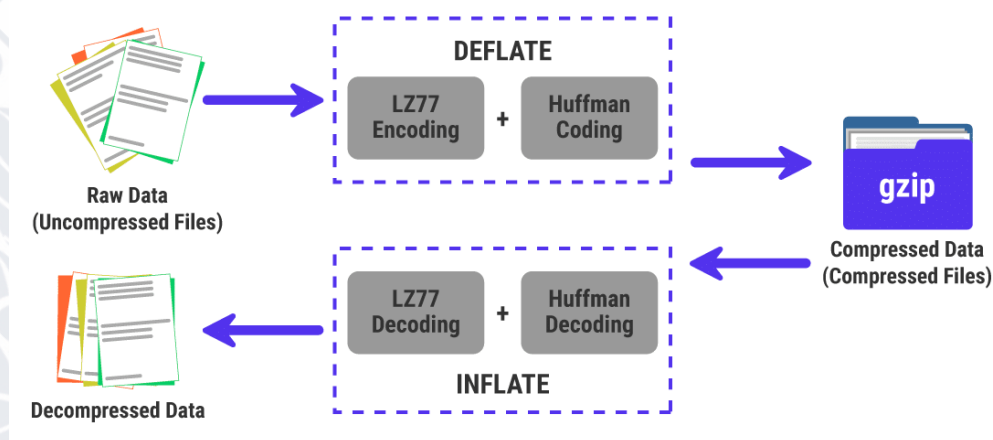


Рисунок 2.7 Принцип стиснення через GZIP

Принцип стиснення полягає в поєднанні двох основних методів стиснення: **LZ77** та **Huffman Coding**. **LZ77** зосереджується на виявленні та заміні повторюваних послідовностей символів. Якщо алгоритм знаходить повторювану послідовність, він замінює її посиланням на попереднє її входження. Замість копіювання однакових даних багато разів, **LZ77** записує тільки позицію та довжину повторення відносно її першого входження. Після того як **LZ77** зменшить обсяг даних шляхом усунення повторень, **Huffman Coding** оптимізує

спосіб, яким ці дані подальше кодуються. **Huffman Coding** аналізує частоту кожного символу у тексті та призначає найкоротші коди найчастіше використовуваним символам. Менш часті символи отримують довші коди.

8. Заради пришвидшення авторизації на сайт та безпечного користування можна застосувати технологію OAuth 2.0 - протокол делегування авторизації, який дозволяє додаткам отримувати обмежений доступ до облікових записів користувача на інших сервісах. Це особливо корисно для надання доступу без необхідності розкривати паролі користувачів. Протокол використовується багатьма великими компаніями, такими як Google, Facebook, Microsoft і Twitter, для управління дозволами і дозволяє користувачам легко і безпечно надавати додаткам доступ до своїх інформації. OAuth 2.0 поділяє процес авторизації на 3 компоненти: **Безпосередня авторизація** – коли користувач надає додатку дозвіл на доступ до своїх даних на сервері ресурсів. **Видача токена** - Після отримання дозволу від користувача, клієнт звертається до сервера авторизації з відповідним запитом на отримання токена. **Використання токена** - Клієнт використовує отриманий токен для доступу до захищених ресурсів на сервері ресурсів. Токен діє як доказ авторизації користувача. Їх буває два види: **Access Token** - Використовується для доступу до захищених ресурсів та є короткотривалим. **Refresh Token** - Використовується для поновлення access token у випадку перезавантаження сторінки, випадкового закриття браузеру, крашу тощо.

OAuth 2.0

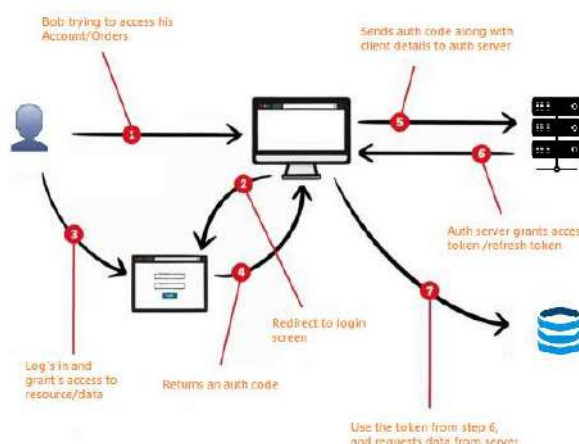


Рисунок 2.8 Принцип роботи OAuth 2.0

9. Використання мікросервісної архітектури

Мікросервісна архітектура є підходом до розробки програмного забезпечення, при якому система розділяється на невеликі, незалежні сервіси, які взаємодіють один з одним через стандартизовані інтерфейси. Вона суттєво відрізняється від застарілої монолітної архітектури, де вся система є єдиною великою програмою.

Кожен мікросервіс розробляється, розгортається та масштабується незалежно від інших. Вони можуть бути написані на різних мовах програмування і використовувати різні технології. Кожен мікросервіс виконує одну чітко визначену функцію або бізнес-логіку і взаємодіє з іншими через стандартизовані інтерфейси, часто через HTTP/HTTPS або через систему повідомлень.

Кожен мікросервіс має свою власну базу даних або сховище даних, що дозволяє уникнути глобальних змін у базах даних і забезпечити автономність сервісів. Команди розробників можуть працювати незалежно одна від одної над різними мікросервісами, тим самим прискорюючи процес розробки і дозволяючи швидше виводити нові функції на ринок.

Основними перевагами мікросервісної архітектури є її масштабованість, стійкість до збоїв, гнучкість у виборі технологій і швидкий випуск нових функцій. Кожен мікросервіс можна масштабувати незалежно від інших, залежно від навантаження. Помилка в одному мікросервісі не впливає на роботу всієї системи. Мікросервіси можуть використовувати найбільш підходящі для них технології та інструменти. Можливість незалежного розгортання дозволяє швидше впроваджувати нові функції.

Однак, мікросервісна архітектура має свої виклики. Велика кількість мікросервісів ускладнює управління ними, в тому числі оркестрацію та моніторинг. Комунікація між мікросервісами здійснюється через мережу, що може призводити до додаткових затримок і витрат на мережеву взаємодію. Більша кількість точок входу підвищує ризики безпеки, тому важливо ретельно контролювати доступ і захищати дані. Розподілені системи ускладнюють управління транзакціями, особливо якщо вони охоплюють кілька мікросервісів.

Приклади використання мікросервісної архітектури включають Netflix та Amazon. Netflix використовує мікросервісну архітектуру для надання своїх потокових сервісів, де кожна функція, від рекомендацій до обробки платежів, реалізована як окремий мікросервіс. Amazon перейшов від моноліту до мікросервісів, що дозволяє йому масштабувати кожну функцію окремо, включаючи каталоги товарів, обробку замовлень і систему оплати.

Для реалізації мікросервісної архітектури використовуються різні інструменти та технології, такі як Docker для контейнеризації мікросервісів, Kubernetes для оркестрації контейнеризованих додатків, Istio для управління мікросервісами та Spring Boot для створення мікросервісів на Java.

Мікросервісна архітектура пропонує значні переваги в масштабованості, гнучкості та стійкості до збоїв, але вона також вимагає ретельного планування і управління для подолання її складностей. Цей підхід особливо корисний для великих, складних систем, які потребують постійного розвитку і масштабування. У порівнянні з монолітною архітектурою, мікросервіси забезпечують кращу масштабованість та гнучкість. Монолітна архітектура передбачає, що вся

система є одним цілим, де всі компоненти тісно інтегровані один з одним. Це може призводити до того, що будь-які зміни в системі вимагають перекомпіляції і розгортання всієї програми, що може бути складним і тривалим процесом. Монолітні додатки часто важко масштабувати, оскільки масштабування означає збільшення всієї програми, а не окремих її частин. Відповідно, помилка в одному компоненті може вплинути на роботу всієї системи. Мікросервісна архітектура вирішує ці проблеми, дозволяючи розробникам працювати з окремими сервісами, які можуть бути розгорнуті та масштабовані незалежно, забезпечуючи більшу гнучкість та стійкість системи.

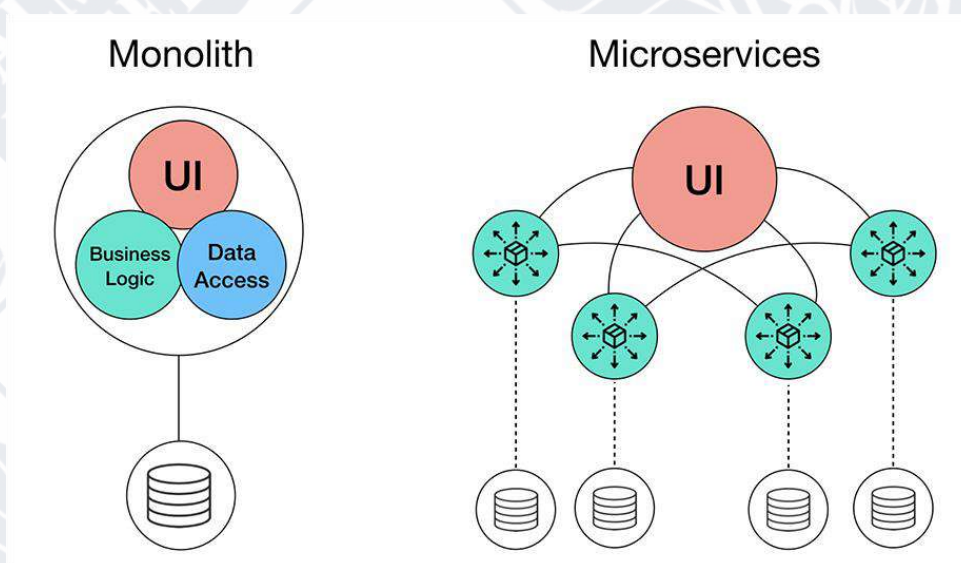


Рисунок 2.9 Схема роботи мікросервісної архітектури в порівнянні з монолітною

Висновок до розділу 2

В другому розділі наведено список технологій та інструментів які будуть використані під час розробки серверної архітектури для сайту надання психотерапевтичних послуг. Друга частина другого розділу містить в собі найвідоміші та найефективніші прийоми для оптимізації роботи веб-ресурсу, які стануть в нагоді при розробці власного продукту.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Опис бази даних

Весь вміст сайту (облікові записи, дати запису, платіжні дані тощо) зберігаються в базі даних. Концептуальна база даних виглядає наступним чином:

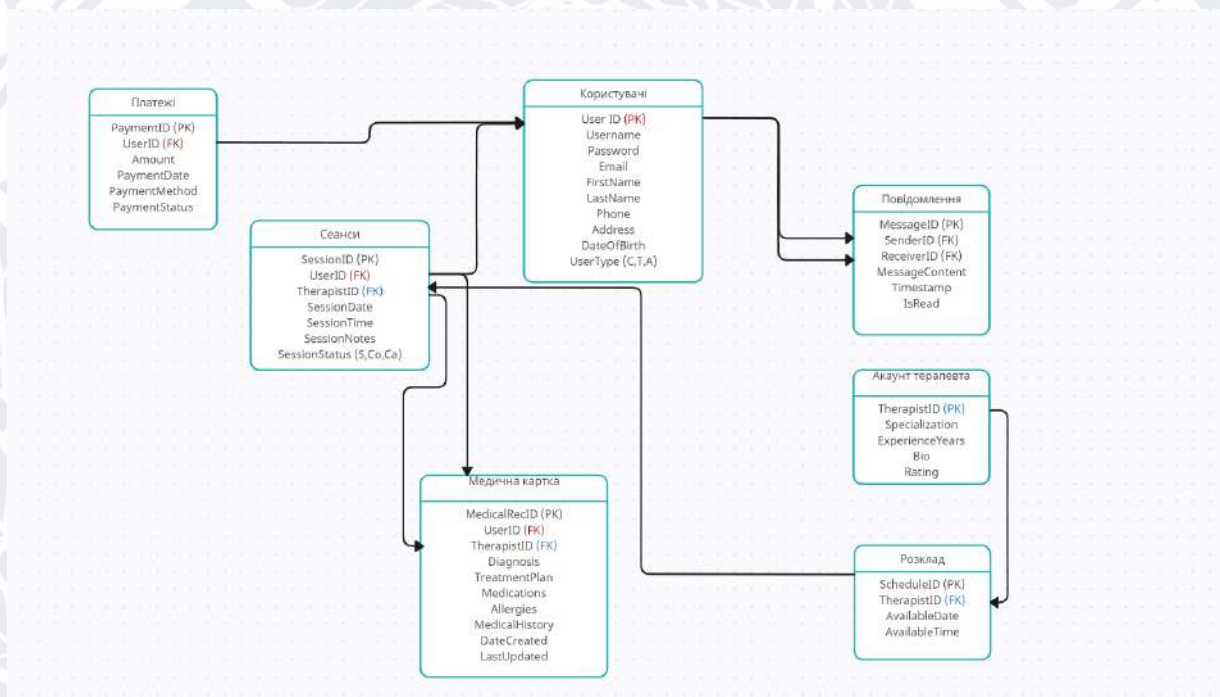


Рис. 3.1 Концептуальна модель бази даних

База даних складається з наступних таблиць: «Користувачі», «Платежі», «Сеанси», «Медична картка», «Повідомлення», «Акаунт терапевта» та «Розклад».

Структура таблиці «Користувачі» наведена нижче:

Назва поля	Тип	Опис
UserID	INT	Унікальний номер користувача
Username	varchar (50)	облікова назва користувача

Password	varchar (255)	пароль для входу в обліковий запис
Email	varchar (100)	Поштовик користувача
FirstName	varchar (50)	Ім'я
LastName	varchar (50)	Прізвище
Phone	varchar (15)	Номер телефону
Address	varchar (255)	Адреса проживання
DateOfBirth	DATE	Дата народження
UserType	ENUM	Унікальний ідентифікатор типу акаунту (C – Client, T – Therapist, A - Admin)

Таблиця «Користувачі» є в цій БД основною, з нею пов'язані більшість таблиць: зв'язок з таблицею «Сеанси» побудований через поле «UserID», з таблицею «Повідомлення» зв'язок через поле «UserID», «ReceiverID» та «SenderID», так як будь-який користувач, що будь-яким чином взаємодіє з повідомленням, може бути як його відправником, так і отримувачем.

Поле «UserID» кожного користувача також пов'язане з полем «UserID» таблиці «Платежі», адже кожен проведений на сайті платіж прив'язаний до того, хто його сплатив використовуючи унікальний ідентифікатор UserID.

Медична картка зв'язується з «Користувачами» через «UserID» лише в тому випадку, коли тип акаунту «Client» або «Admin» (Медична картка адмінів необхідна з метою тестування продукту).

Будова таблиці «Повідомлення»:

Назва поля	Тип	Опис
MessageId	INT	Унікальний ідентифікатор повідомлення

SenderID	INT	ідентифікатор відправника повідомлення, який являє собою UserID
ReceiverID	INT	ідентифікатор отримувача повідомлення, UserID
MessageContent	TEXT	Вміст повідомлення
Timestamp	DATETIME	Маркер часу
IsRead	BOOLEAN	Маркер прочитаного повідомлення

Будова таблиці «Платежі»

Назва поля	Тип	Опис
PaymentID	INT	ідентифікатор платежу
UserID	INT	номер користувача
Amount	DECIMAL (10,2)	Сума платежу
PaymentDate	DATE	Дата оплати
PaymentMethod	varchar (50)	Метод оплати
PaymentStatus	ENUM	Статус оплати (P – Pending, C – Completed, F – Failed)

Будова таблиці «Акаунт терапевта»

Назва поля	Тип	Опис
TherapistID	INT	Унікальний ідентифікатор спеціаліста, являє собою UserID
Specialization	varchar(100)	Спеціалізація
ExperienceYears	INT	Досвід

Bio	TEXT	Коротка біографія, що віображається в особистому портфоліо
Rating	DECIMAL(3,2)	Рейтинг спеціаліста

«Акаунт терапевта» є модифікацією таблиці «Користувачі» у випадку якщо тип акаунту «Therapist». Цей тип наділяє власника акаунту додатковими можливостями, на кшталт особистого кабінету спеціаліста, особистого гаманця та адміністрування чату з потенційним пацієнтом. При цьому такий тип акаунта не має доступу до запису на сеанс як пацієнт.

Таблиця «Розклад»

Назва поля	Тип	Опис
ScheduleID	INT	Унікальний ідентифікатор розкладу
TherapistID	INT	Ідентифікатор терапевта
AvailableDate	DATE	Доступна дата
AvailableTime	TIME	Доступний час

Таблиця «Розклад» зв'язана з таблицею «Користувачі» через поле «UserID» та з таблицею «Акаунт терапевта» через поле «TherapistID». Так як унікальний ідентифікатор користувача та терапевта це одне й те ж саме, різниця лише в типі акаунту. Розклад прив'язаний до кожного акаунту типу «Терапевт».

Будова таблиці «Медична картка»

Назва поля	Тип	Опис
MedicalRecordID	INT	Унікальний ідентифікатор медичної картки, зв'язаний з UserID
UserID	INT	Ідентифікатор користувача

TherapistID	INT	Ідентифікатор терапевта, який закріплений за пацієнтом в медичній картці
Diagnosis	TEXT	Діагноз
TreatmentPlan	TEXT	План лікування захворювання при визначеному діагнозі
Medications	TEXT	Призначені медикаменти
Allergies	TEXT	Наявні алергії
MedicalHistory	TEXT	Історія хвороб
DateCreated	DATE	Дата створення
LastUpdated	DATETIME	Останнє оновлення даних

Таблиця «Медична картка» пов'язана з таблицею «Користувачі» через поле «UserId». Таким чином, якщо тип акаунту користувача Client йому автоматично при успішній реєстрації присвоює медичну картку закріплену за його номером акаунту, який генерується також в момент успішної реєстрації і залишається незмінним задля забезпечення цілісності даних в базі. Зв'язок з таблицею «Сеанси» відбувається через поле «TherapistId». В таблиці «Медична картка» записуються всі спеціалісти, з якими пацієнт проходив лікування. Відповідно, через унікальний ідентифікатор лікаря сеанс терапії підв'язується в медичну картку.

Таблиця «Сеанси»

Назва поля	Тип	Опис
SessionId	INT	Унікальний ідентифікатор сеансу
UserId	INT	Унікальний ідентифікатор пацієнта
TherapistId	INT	Унікальний ідентифікатор терапевта
SessionDate	DATE	Дата сеансу
SessionTime	TIME	Час сеансу

SessionNote	TEXT	Нотатки сеансу
SessionStatus	ENUM	Статус сеансу (S – Scheduled, Co – Completed, Ca - Cancelled)

3.2 Програмна реалізація

Для реалізації форми реєстрації необхідно потурбуватися про створення маршрутів та контролерів, включно з маршрутами для обробки запитів на реєстрацію, перевірку даних і збереження нових користувачів у базу даних. Важливо впровадити валідацію введених даних, щоб перевіряти їх правильність (формат email, довжина паролю) і обробляти помилки валідації з відповідними повідомленнями для користувача.

Крім того, необхідно реалізувати шифрування паролів перед їх збереженням у базу даних. Завершивши основну розробку, слід переконатися в справній роботі форми, внесення даних в базу даних та можливості успішного входу на сайт за обліковими даними.

Для написання коду для написання бекенду форми реєстрації скористаємось RНР:

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];

    if ($password != $confirm_password) {
        echo "Паролі не співпадають.";
        exit();
    }

    $servername = "localhost";
    $db_username = "root";
    $db_password = "";
    $dbname = "registration_db";

    $conn = new mysqli($servername, $db_username, $db_password, $dbname);

    if ($conn->connect_error) {
        die("Помилка підключення: " . $conn->connect_error);
    }

    $sql = "INSERT INTO users (username, email, password) VALUES ('$username', '$email', '$password')";

    if ($conn->query($sql) === TRUE) {
        echo "Реєстрація успішна!";
    } else {
        echo "Помилка: " . $sql . "<br>" . $conn->error;
    }

    $conn->close();
}

```

Рис. 3.2 Лендинг коду форми реєстрації

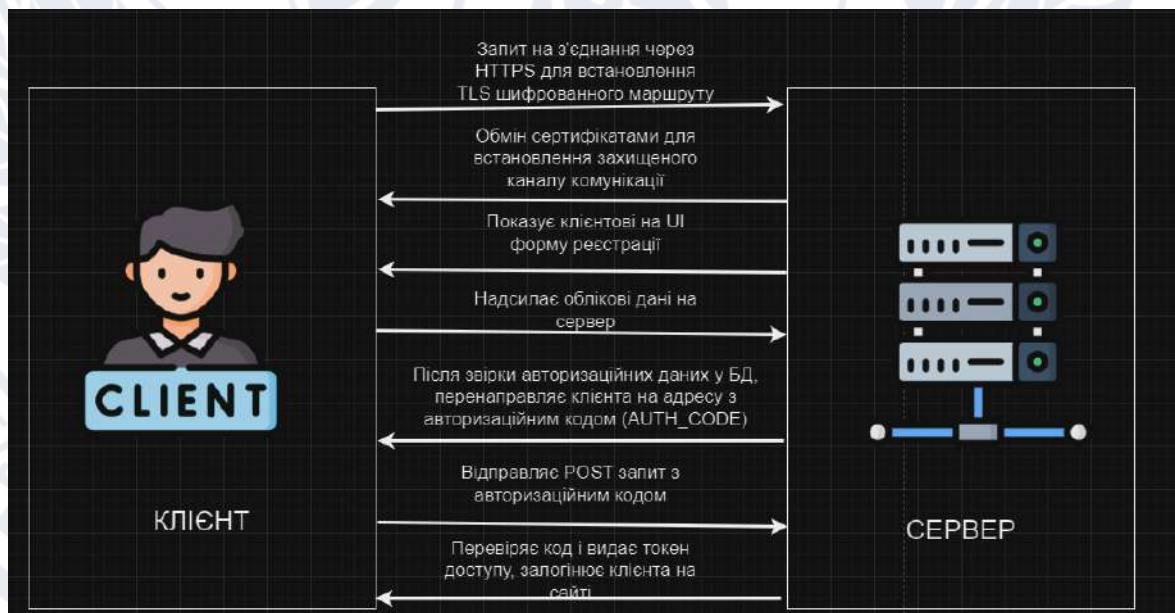


Рис. 3.3 Схема роботи реєстрації на сайті

Після успішної реєстрації на сайті, облікові дані (включаючи логін та пароль) зберігаються в базі даних. Потім щоб зайти під своїм акаунтом

користувачеві потрібно ввести ці ж дані у вікні реєстрації, і якщо дані співпадуть з наявними в базі, процедура видачі токена доступу на сайт повторюється, і клієнт залогінується.

Схема запису на прийом, що імплементується на сайті, працює наступним чином:

1. Користувач заповнює облікові дані.
2. Обирає тип терапії.
3. Вибір діагнозу (якщо встановлений)
4. Після вибору терапії (та діагнозу) коло доступних йому спеціалістів звужується, тож пацієнтові потрібно обрати собі лікаря.
5. Вибір тривалості процедури (45 хв, 1 год, 1.5 год, 3 год)
6. Форма запису вираховує вартість процедури за наступною формулою:
$$\text{Вартість} = \text{БазоваВартість} * \text{Досвід} * (\text{ТривалістьГод}) + \text{СервіснийЗбір}$$

Базова вартість становить 250 гривень. Досвід враховується в роках, округлений до 0.5 в меншу сторону (Наприклад, якщо в спеціаліста 1.7 років досвіду, його коефіцієнт досвіду у формулі буде 1.5)
7. Вибір дати процедури у невеличкому віджеті-календарі.
8. Далі слідує додаткова перевірка системою, що терапевт вільний в обраний час і підтверджує запис на сеанс. У випадку якщо терапевт зайнятий, система видасть відповідну помилку.
9. Якщо запис проходить без помилок, система зберігає інформацію про запис, генерується посилання на майбутню Zoom-зустріч.

```
// Функція для підключення до бази даних
function get_db_connection() {
    $host = 'localhost'; // Ім'я хосту
    $db = 'database'; // Ім'я бази даних
    $user = 'username'; // Ім'я користувача
    $pass = 'password'; // Пароль

    $dsn = "mysql:host=$host;dbname=$db;charset=utf8";
    try {
        $pdo = new PDO($dsn, $user, $pass);
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        return $pdo;
    } catch (PDOException $e) {
        echo 'Підключення не вдалося: ' . $e->getMessage();
        exit;
    }
}

// Функція для отримання спеціалістів з бази даних
function fetch_specialists() {
    $pdo = get_db_connection();
    $stmt = $pdo->query('SELECT id, name FROM specialists');
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}
```

Рис. 3.4 Функція отримання спеціаліста з БД

```
function process_form() {
    $specialist_id = $_POST['specialist'];
    $session_type = $_POST['session_type'];
    $session_duration = $_POST['session_duration'];
    $name = $_POST['name'];
    $age = $_POST['age'];
    $gender = $_POST['gender'];
    $diagnosis = $_POST['diagnosis'];
    $date = $_POST['date'];

    // Отримання імені спеціаліста з бази даних
    $pdo = get_db_connection();
    $stmt = $pdo->prepare('SELECT name FROM specialists WHERE id = ?');
    $stmt->execute([$specialist_id]);
    $specialist = $stmt->fetchColumn();
}
```

Рис. 3.5 Запит до БД для отримання спеціаліста

Після проведення сеансу терапії, користувач матиме змогу скористатися формою зворотнього зв'язку:

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Отримання даних з форми
    $name = htmlspecialchars($_POST['name']);
    $email = htmlspecialchars($_POST['email']);
    $rating = intval($_POST['rating']);
    $comments = htmlspecialchars($_POST['comments']);

    // Валідація даних
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo "Некоректна електронна адреса";
        exit;
    }

    if ($rating < 1 || $rating > 5) {
        echo "Оцінка повинна бути між 1 та 5";
        exit;
    }

    // Збереження даних
    $feedback = "Ім'я: $name\nЕлектронна пошта: $email\nОцінка: $rating\nКоментар: $comments\n\n";
    file_put_contents('feedback.xml', $feedback, FILE_APPEND);

    echo "Дякуємо за ваш зворотній зв'язок!";
}
```

Рис. 3.6 Функціонал форми зворотнього зв'язку

```
function process_form() {
    $specialist_id = $_POST['specialist'];
    $session_type = $_POST['session_type'];
    $session_duration = $_POST['session_duration'];
    $name = $_POST['name'];
    $age = $_POST['age'];
    $gender = $_POST['gender'];
    $diagnosis = $_POST['diagnosis'];
    $date = $_POST['date'];
}
```

Рис. 3.7 Збереження даних про сеанс в БД

Висновок до розділу 3

В цьому розділі було наведено програмну реалізацію продукту. Створено концептуальну модель бази даних з поясненнями ролі кожної таблиці. Також було пояснено принцип роботи ключових компонентів системи з фрагментами коду.

ВИСНОВКИ

По завершенню роботи було успішно виконано початково поставлені задачі. Досліджено актуальність платформи психотерапевтичних послуг. Розглянуто вітчизняні та світові аналоги, досліджено проблематику потенційних конкурентів та тематики розробки подібного ресурсу в цілому. В ході огляду аналогів продукту було виявлено нестачу оптимізації у вітчизняних сайтах з надання психотерапевтичних та психологічних послуг. Також було обгрунтовано необхідність в розробці оптимізованого, швидкого та перспективного бекенду для власного веб-продукту з метою зробити внесок в Українське інтернет-суспільство турботи про ментальне здоров'я.

Для розробки серверної архітектури були досліджені достатньо методів, інструментів та технологій для коректної розробки та оптимізації. Було обрано такі технології як PHP, PHPMyAdmin та MongoDB. В ролі інструментів оптимізації роботи продукту були обрані наступні прийоми: нормалізація БД, індексація ключових полів БД, кешування часто використовуваних запитів, CSS Sprites, CDN, та OAuth 2.0.

В результаті завершального мануального тестування було знайдено та виправлено кілька неточностей в коді. Набуто вдосталь навичок по імплементації найновіших технологій розробки та оптимізації, які використовують багато інтернет гігантів по всьому світі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нормалізація схеми Баз даних [Електронний ресурс] – Доступ до ресурсу: https://bazidanih5.blogspot.com/p/blog-page_74.html
2. CSS Image Sprites [Електронний ресурс] – Доступ до ресурсу: <https://www.formget.com/css-image-sprites/>
3. What Is a CDN and How Do CDNs Work? [Електронний ресурс] – Доступ до ресурсу: <https://www.shopify.com/ph/blog/what-is-cdn>
4. How to Enable GZIP Compression to Speed Up WordPress Sites [Електронний ресурс] – Доступ до ресурсу: <https://kinsta.com/blog/enable-gzip-compression/>
5. OAuth 2.0 Understanding Step-by-Step Guide [Електронний ресурс] – Доступ до ресурсу: <https://medium.com/codenx/oauth-2-0-4cddd6c7471f>
6. Online Therapy for Everyone [Електронний ресурс] – Доступ до ресурсу: <https://www.talkspace.com/online-therapy>
7. Talkiatry - Psychiatry, with you in mind [Електронний ресурс] – Доступ до ресурсу: <https://www.talkiatry.com>
8. Convenient and affordable therapy with BetterHelp. [Електронний ресурс] – Доступ до ресурсу: <https://www.betterhelp.com/get-started/>
9. Підберіть психотерапевта, який вас зрозуміє [Електронний ресурс] – Доступ до ресурсу: <https://www.rozmova.me>
10. Захворів? Нема лікаря поруч? – E-Likari [Електронний ресурс] – Доступ до ресурсу: <https://e-likari.com.ua>
11. pleso - психотерапія онлайн [Електронний ресурс] – Доступ до ресурсу: <https://pleso.me/ua>
12. Мікросервісний підхід до веб розробки [Електронний ресурс] – Доступ до ресурсу: <https://dou.ua/lenta/articles/micro-frontend/>