

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Відокремлений структурний підрозділ
«Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»
Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

**на тему: Розробка дизайну та навігаційної структури вебсайту для
психолога Ольги Демедчук**

Виконав: студент IV курсу, групи КН-421

спеціальності: 122 Комп'ютерні науки

Мацюк Дмитро

Керівник

Галина МАРЦІЯШ

Рецензент

(ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук
Освітньо-професійний ступінь «фаховий молодший бакалавр»
Спеціальність 122 Комп'ютерні науки
Галузь знань 12 Інформаційні технології

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук

_____ Галина МАРЦІЯШ

« 02 » березня 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Мацюку Дмитру Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка дизайну та навігаційної структури вебсайту для психолога
Ольги Демедчук

керівник роботи Марціяш Галина Ярославівна,

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студенткою роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення,
мови програмування: JavaScript; фреймворки та бібліотеки: React, Vite, Node.js,
Express.js, Tailwind CSS; база даних: MongoDB Atlas, Mongoose; засоби авторизації:
JWT, bcrypt; зовнішні сервіси: Cloudinary, Vercel, Render., стандарти IEEE 29148-2018,
IEEE 29119, ДСТУ 8302:2015.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- 1 Загальний розділ
 - 1.1 Аналітичний огляд існуючих рішень
 - 1.2 Технічне завдання
 - 1.2.1 Найменування та область застосування
 - 1.2.2 Призначення розробки
 - 1.2.3 Вимоги до функціоналу вебзастосунку
 - 1.2.4 Вимоги до програмної документації
 - 1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

2 Розробка технічного та робочого проекту

2.1 Розробка структури сайту і вебсторінок

2.2 Створення та верстка сторінок сайту

2.3 Розробка структури бази даних сайту

2.4 Програмування сайту

2.4.1 Написання клієнтської частини

2.4.2 Написання admin частини

2.5 Тестування вебсайту

3 Спеціальний розділ

3.1 Розміщення сайту в Інтернеті

3.2 Інструкція з обслуговування та наповнення сайту

3.3 Інструкція з популяризації та підтримки сайту

4 Економічний розділ

4.1 Визначення стадій технологічного процесу та загальної тривалості

проведення НДР

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

4.3 Розрахунок витрат на електроенергію

4.4 Розрахунок суми амортизаційних відрахувань веб-сайту для психолога

Ольги Демедчук

4.5 Обчислення накладних витрат

4.6 Складання кошторису витрат та визначення собівартості веб-сайту для

психолога Ольги Демедчук

4.7 Розрахунок ціни для веб-сайту для психолога Ольги Демедчук

4.8 Визначення економічної ефективності і терміну окупності капітальних

вкладень

5 Охорона праці, техніка безпеки та екологічні вимоги

5.1 Інструкції з охорони праці, їх поділ

5.2 Електробезпека. Види ураження людини електричним струмом

6 Висновки

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – вебсайту.

5. Перелік графічного матеріалу:

1. UML-діаграма варіантів використання програми

2. Структурна схема головної сторінки сайту

3. ER-діаграма бази даних вебсайту

4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Генадій ГОРЯЧЕК		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	20.03.2026
2	Збір і узагальнення інформації	01.05.2026	01.05.2026
3	Написання першого розділу	15.05.2026	15.05.2026
4	Розробка технічного та робочого проєкту	29.05.2026	29.05.2026
5	Написання спеціального розділу	05.06.2026	05.06.2026
6	Розрахунок економічної частини	08.06.2026	08.06.2026
7	Написання розділу охорони праці	09.06.2026	09.06.2026
8	Виконання графічної частини	10.06.2026	10.06.2026
9	Оформлення пояснювальної записки	11.06.2026	11.06.2026
10	Погодження нормоконтролю	12.06.2026	12.06.2026
11	Попередній захист кваліфікаційної роботи	16.06.2026	16.06.2026
12	Захист кваліфікаційної роботи	22.06.2026	22.06.2026

7. Дата видачі завдання: 05 березня 2026 р.

Студент

(підпис)

Дмитро МАЦЮК

Керівник роботи

(підпис)

Галина МАРЦІЯШ

ЗМІСТ

АНОТАЦІЯ	7
ВСТУП	9
1 ЗАГАЛЬНИЙ РОЗДІЛ	10
1.1 Аналітичний огляд існуючих рішень	10
1.2 Технічне завдання	11
1.2.1 Найменування та область застосування	12
1.2.2 Призначення розробки	13
1.2.3 Вимоги до функціоналу вебсайту	13
1.2.4 Вимоги до програмної документації	16
1.2.5 Техніко-економічні показники	17
1.2.6 Стадії та етапи розробки	19
1.2.7 Порядок тестування та прийому	20
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ	22
2.1 Розробка структури сайту і вебсторінок	22
2.2 Створення та верстка сторінок сайту	24
2.3 Розробка структури бази даних сайту	28
2.4 Програмування сайту	32
2.4.1 Написання клієнтської частини	35
2.4.2 Написання адміністративної частини	36
2.5 Тестування вебсайту	38
3 СПЕЦІАЛЬНИЙ РОЗДІЛ	43
3.1 Розміщення сайту в Інтернеті	43
3.2 Інструкція з обслуговування та наповнення сайту	47
3.3 Інструкція з популяризації та підтримки сайту	52

					2026.KBP.122.421.15.00.00 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Мацюк Д. В.			«Розробка дизайну та навігаційної структури вебсайту для психолога Ольги Демедчук» Пояснювальна записка	Літ.	Арк.	Аркуші
Перевір.		Марціаш Г. Я.					5	114
Реценз.						ВСП ТФК ТНТУ КН-421 м. Тернопіль		
Н. Контр.		Приймак В. А.						
Затверд.								

4	ЕКОНОМІЧНИЙ РОЗДІЛ.....	58
4.1	Визначення стадій технологічного процесу та загальної тривалості проведення НДР	58
4.2	Визначення витрат на оплату праці та відрахувань на соціальні заходи60	
4.3	Розрахунок витрат на електроенергію	61
4.4	Розрахунок суми амортизаційних відрахувань вебсайту для психолога Ольги Демедчук.....	62
4.5	Обчислення накладних витрат	63
4.6	Складання кошторису витрат та визначення собівартості вебсайту для психолога Ольги Демедчук.....	63
4.7	Розрахунок ціни для вебсайту для психолога Ольги Демедчук.....	64
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень	64
5	ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ66	
5.1	Інструкції з охорони праці, їх поділ	66
5.2	Електробезпека. Види ураження людини електричним струмом.....	68
	ВИСНОВКИ.....	71
	ПЕРЕЛІК ПОСИЛАНЬ	72
	ДОДАТКИ.....	73
	Додаток А. App.jsx	73
	Додаток Б. navigation.js.....	76
	Додаток В. HomePage.jsx.....	77
	Додаток Г. ServicesPage.jsx	78
	Додаток Д. BookingPage.jsx.....	79
	Додаток Е. CourseDetailPage.jsx	84
	Додаток Ж. payloadClient.js.....	91
	Додаток И. server.js	93
	Додаток К. courseRoutes.js.....	97
	Додаток Л. admin.js	98

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Розробка дизайну та навігаційної структури вебсайту для психолога Ольги Демедчук».

Метою кваліфікаційної роботи є розробка сучасного адаптивного вебсайту для психолога, який забезпечує зручне представлення професійної діяльності спеціаліста, продуману навігаційну структуру, зрозумілий користувацький інтерфейс, особистий кабінет користувача та адміністративну панель для керування контентом. Пояснювальна записка містить 5 розділів.

У загальній частині подано аналітичний огляд існуючих рішень, визначено призначення розробки, область застосування вебсайту, основні вимоги до функціоналу, програмної документації, техніко-економічних показників, стадії розробки та порядок тестування з урахуванням виявлених потреб.

У другому розділі описано процес розробки технічного та робочого проєктів. Розглянуто структуру вебсайту і вебсторінок, створення та верстку сторінок, розробку бази даних, програмування клієнтської та серверної частин, реалізацію адміністративної панелі та тестування вебсайту для забезпечення його функціонування.

У спеціальній частині розглянуто порядок розміщення сайту в мережі Інтернет, налаштування frontend- і backend-частин, підключення MongoDB Atlas, використання зовнішніх сервісів, інструкцію з обслуговування та наповнення сайту, а також способи його популяризації й підтримки для подальшої роботи.

Розрахунок вартості розробки та економічної ефективності наведено в економічній частині. Основні питання охорони праці, техніки безпеки та екологічних вимог розглянуто в п'ятому розділі.

Обсяг пояснювальної записки становить 107 сторінок.

До складу кваліфікаційної роботи входить графічна частина, яка містить структурну схему вебсайту, ER-діаграму бази даних, UML-діаграми, схеми основних процесів роботи сайту, техніко-економічні показники, виконані на окремих аркушах формату А1.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

ANNOTATION

The topic of the qualification work is “Development of the design and navigation structure of a website for psychologist Olha Demedchuk”.

The purpose of the qualification work is to develop a modern adaptive website for a psychologist, which provides a convenient presentation of the specialist’s professional activity, a well-designed navigation structure, a clear user interface, a personal user account, and an administrative panel for content management. The explanatory note consists of 5 sections.

The general section presents an analytical review of existing solutions, defines the purpose of the development, the field of application of the website, the main requirements for functionality, software documentation, technical and economic indicators, development stages, and the testing procedure, taking into account the identified needs.

The second section describes the process of developing the technical and working projects. It considers the structure of the website and web pages, the creation and layout of pages, database development, programming of the client and server parts, implementation of the administrative panel, and website testing to ensure its proper functioning.

The special section considers the procedure for deploying the website on the Internet, configuring the frontend and backend parts, connecting MongoDB Atlas, using external services, instructions for maintaining and filling the website with content, as well as methods of its promotion and support for further operation.

The calculation of the development cost and economic efficiency is presented in the economic section. The main issues of occupational safety, safety engineering, and environmental requirements are considered in the fifth section.

The explanatory note consists of 107 pages.

The qualification work includes a graphical part containing the structural scheme of the website, the ER diagram of the database, UML diagrams, schemes of the main website operation processes, and technical and economic indicators, presented on separate A1 format sheets.

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

У сучасних умовах цифрові технології відіграють важливу роль у розвитку сфери психологічних послуг. Все більше користувачів шукають в Інтернеті інформацію про спеціалістів, консультації, курси та онлайн-заходи. Власний вебсайт для приватного психолога є важливим інструментом для представлення професійної діяльності, взаємодії з клієнтами та формування довіри.

Актуальність розробки полягає в тому, що використання лише месенджерів, соціальних мереж або сторонніх сервісів не завжди забезпечує зручну роботу з інформацією. Дані про послуги, курси, вебінари, відгуки та контакти можуть бути розміщені в різних джерелах, що ускладнює їх пошук і сприйняття. Тому єдиний веб-ресурс дозволяє впорядкувати інформацію та зробити взаємодію з користувачами зрозумілішою.

Сучасний вебсайт дає змогу об'єднати основні напрями діяльності спеціаліста в одному середовищі. Завдяки продуманій навігаційній структурі користувач може швидко перейти до потрібного розділу, ознайомитися з послугами, переглянути курси, зареєструватися на вебінар, прочитати відгуки або скористатися особистим кабінетом. Таким чином, адаптивний інтерфейс забезпечує коректне відображення сайту на різних пристроях і підтримує зручну роботу з інформацією.

Метою кваліфікаційної роботи є розробка дизайну та навігаційної структури вебсайту для психолога Ольги Демедчук. У межах роботи передбачено створення публічної частини сайту, сторінок послуг, курсів, вебінарів, відгуків і контактів, а також особистого кабінету та адміністративної панелі для керування контентом.

Реалізація такого вебсайту дозволяє централізувати інформацію про діяльність психолога, зробити взаємодію з користувачами зручнішою та забезпечити подальший розвиток ресурсу. Отже, розроблений проєкт може використовуватися для презентації професійної діяльності, підтримки курсів і вебінарів, обробки заявок та організації онлайн-присутності психолога в мережі Інтернет.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

У сучасних умовах вебсайт є одним із основних інструментів для представлення діяльності спеціаліста в мережі Інтернет. Для психолога сайт допомагає сформувати довіру до фахівця, показати напрями роботи та надати користувачам доступ до послуг, навчальних матеріалів, вебінарів і контактної інформації.

Під час аналізу наявних рішень було розглянуто кілька типів вебсайтів, які можуть використовуватися психологами та іншими спеціалістами у сфері консультивання. До них належать персональні сайти, сайти-візитки, сайти послуг, онлайн-платформи для навчання, сторінки з вебінарами та сайти з особистим кабінетом користувача.

Найпростішим варіантом є сайт-візитка. Такий сайт зазвичай містить коротку інформацію про спеціаліста, опис послуг, контактні дані та форму зворотного зв'язку. Перевагою такого рішення є простота реалізації та швидке завантаження сторінок. Проте сайт-візитка має обмежені можливості, тому не вирішує потреби у повноцінному керуванні курсами, вебінарами, користувачами, заявками та навчальними матеріалами.

Іншим поширеним рішенням є сайт послуг. Він має ширшу структуру ніж сайт-візитка і дозволяє детальніше описати напрями роботи спеціаліста. Такий сайт може містити окремі сторінки для послуг, відгуків, контактів і запису на консультацію. Однак у більшості випадків він також не передбачає особистого кабінету користувача та навчальної частини.

Окрему групу становлять навчальні платформи або LMS-системи. Вони дозволяють користувачам проходити курси, переглядати уроки, відстежувати прогрес і отримувати доступ до матеріалів після оплати. Такі рішення є функціональними, але часто мають складний інтерфейс і не завжди підходять для персонального бренду окремого спеціаліста. Крім того, готові LMS-платформи

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

можуть мати обмеження щодо дизайну, структури сторінок і способів подання інформації, що ускладнює створення потрібного рішення.

Також існують окремі сервіси для проведення вебінарів, прийому оплат і запису на консультації. Їх можна використовувати як допоміжні інструменти, але вони не створюють єдиного інформаційного середовища. Користувач змушений переходити між різними платформами, що знижує зручність взаємодії та ускладнює керування даними, тому виникає потреба в об'єднаному рішенні.

Проаналізовані рішення показали, що для теми кваліфікаційної роботи доцільно розробити власний вебсайт для психолога Ольги Демедчук, який поєднує функції персонального сайту, сайту послуг, навчальної частини, модуля вебінарів, особистого кабінету та адміністративної панелі. Такий підхід дозволяє створити єдину навігаційну структуру, адаптувати дизайн під тематику психологічних послуг і забезпечити централізоване керування контентом.

Вебсайт для психолога Ольги Демедчук має перевагу над простим сайтом-візиткою, оскільки поєднує інформаційні сторінки з авторизацією користувачів, переглядом курсів, реєстрацією на вебінари, створенням заявок на оплату та роботою з особистим кабінетом. Адміністративна панель дає змогу керувати вмістом сайту без прямого редагування бази даних або програмного коду, що усуває обмеження простих веб-рішень.

Отже, аналіз наявних рішень підтвердив доцільність створення власного вебсайту для психолога Ольги Демедчук, який відповідає потребам користувача та забезпечує можливості для подальшого розвитку.

1.2 Технічне завдання

Технічне завдання визначає основні вимоги до розробки вебсайту психолога Ольги Демедчук. У ньому описано призначення сайту, область застосування, функціональні можливості, вимоги до надійності, умови експлуатації, склад програмної документації, етапи розробки та порядок тестування.

Структуру технічного завдання та пояснювальної записки сформовано

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

відповідно до методичних вказівок до виконання кваліфікаційної роботи [1]. Розроблений вебсайт повинен забезпечити зручне представлення інформації про психолога, послуги, курси, вебінари, відгуки та контакти, щоб користувачі могли швидко отримати потрібні відомості. Крім цього, сайт має підтримувати реєстрацію користувачів, авторизацію, особистий кабінет, доступ до навчальних матеріалів після підтвердження оплати та адміністративне керування контентом, щоб забезпечити доступ до сервісів і матеріалів на сайті.

1.2.1 Найменування та область застосування

Найменування програмного продукту — вебсайт психолога Ольги Демедчук.

Тема кваліфікаційної роботи — «Розробка дизайну та навігаційної структури вебсайту для психолога Ольги Демедчук».

Розроблений програмний продукт є full-stack вебсайтом, який складається з клієнтської та серверної частин. Клієнтська частина реалізована з використанням React [2], Vite [3], React Router DOM[4] і Tailwind CSS. Серверна частина створена на основі Node.js [5], Express.js [6], MongoDB Atlas [7] і Mongoose [8].

Область застосування сайту — представлення професійної діяльності психолога в мережі Інтернет, інформування відвідувачів про послуги, курси та вебінари, організація доступу до навчальних матеріалів, збір заявок і керування контентом через адміністративну панель, щоб задовольнити потреби відвідувачів, потенційних клієнтів, учасників вебінарів і адміністратора.

Сайт може використовуватися:

- відвідувачами, які хочуть ознайомитися з діяльністю психолога;
- потенційними клієнтами, які шукають послуги психологічної допомоги;
- користувачами, які бажають придбати онлайн-курси; учасниками вебінарів;
- адміністратором, який керує послугами, курсами, вебінарами, відгуками, користувачами, що забезпечує відповідність сайту потребам цих груп.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

1.2.2 Призначення розробки

Експлуатаційне призначення вебсайту полягає у зручному доступі користувачів до інформації про психолога Ольгу Демедчук, її послуги, курси, вебінари, відгуки та контактні дані. Сайт має бути доступним через браузер і працювати на персональних комп'ютерах, ноутбуках, планшетах і смартфонах.

Функціональне призначення розробки полягає у створенні єдиної системи, яка поєднує публічну інформаційну частину, навчальну LMS-частину, модуль вебінарів, особистий кабінет користувача та адміністративну панель. За допомогою сайту користувач може переглядати послуги, відкривати каталог курсів, ознайомлюватися з програмою курсу, створювати заявку на оплату, реєструватися на вебінари та переглядати доступні матеріали після авторизації.

Адміністратор сайту повинен мати можливість керувати основним наповненням: додавати й редагувати послуги, курси, модулі, уроки, вебінари, фотографії, відгуки, користувачів, замовлення та доступи, щоб підтримувати сайт в актуальному стані без прямого втручання в базу даних.

1.2.3 Вимоги до функціоналу вебсайту

Вебсайт повинен виконувати інформаційні, навчальні, користувацькі та адміністративні функції, щоб забезпечити зручну роботу користувачів і адміністрування контенту. Основні вимоги до функціоналу сформульовані з урахуванням поведінки системи з точки зору вхідних і вихідних даних, без деталізації внутрішньої реалізації окремих програмних модулів.

До основних функціональних характеристик сайту належать функції, що стосуються сайту психолога:

- відображення сторінки послуг;
- відображення каталогу онлайн-курсів;
- відкриття детальної сторінки курсу;
- створення заявки на оплату курсу;

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

- авторизація та реєстрація користувача;
- відновлення та скидання пароля;
- відкриття особистого кабінету користувача;
- перегляд доступних курсів;
- проходження навчальних матеріалів;
- відображення сторінки вебінарів;
- реєстрація користувача на вебінар;
- відображення відгуків;
- відображення контактної інформації;
- керування контентом через адміністративну панель;
- підтвердження або відхилення заявок на оплату;
- Завантаження та використання фотографій через Cloudinary [9].

Вхідною інформацією для вебсайту є дані, які вводяться користувачем або адміністратором, а також дані, що зберігаються у базі MongoDB Atlas і стосуються функцій сайту психолога. Ці дані забезпечують роботу основних сторінок, особистого кабінету, адміністративної панелі та доступу до навчальних матеріалів. До вхідних даних належать email, пароль, ім'я, прізвище, телефон, дані профілю, інформація про курси, модулі, уроки, вебінари, послуги, відгуки, замовлення, статуси оплат, посилання на фото, відео, PDF-матеріали та Google Meet.

Вихідною інформацією є дані, які система відображає користувачеві на сторінках сайту психолога. До них належать інформаційні блоки головної сторінки, перелік послуг, список курсів, опис курсу, програма курсу, список вебінарів, відгуки, контактна інформація, повідомлення про успішну або помилкову дію, дані особистого кабінету, навчальні матеріали та статуси доступу. Такий поділ дозволяє зрозуміти, які дані надходять до системи та який результат отримує користувач під час роботи з вебсайтом. Це також спрощує подальше проектування бази даних і визначення зв'язків між основними модулями вебсайту. Організацію вхідної та вихідної інформації наведено в таблиці 1.1.

					<i>2026.KBP.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

Таблиця 1.1 – Вхідна та вихідна інформація вебсайту

Вид інформації	Приклад даних	Джерело або призначення
Дані користувача	email, пароль, ім'я, телефон	Реєстрація, авторизація, профіль
Дані курсів	назва, опис, ціна, програма	Відображення каталогу курсів
Дані вебінарів	назва, дата, ціна, посилання	Відображення вебінарів і реєстрація
Дані послуг	назва, опис, фото	Відображення сторінки послуг
Дані замовлень	користувач, сума, статус	Створення та підтвердження оплати
Дані відгуків	ім'я, текст, оцінка	Відображення та модерація відгуків
Вихідні повідомлення	успіх, помилка, відмова доступу	Інформування користувача про результат дії

Система повинна коректно реагувати на передбачені дії користувача, щоб запобігти несанкціонованому доступу та забезпечити доступ до дозволеного контенту. Якщо користувач не авторизований і намагається придбати курс або відкрити захищену сторінку, система повинна перенаправити його на сторінку входу. Якщо користувач має підтверджений доступ до курсу, система повинна дозволити перегляд навчальних матеріалів. Якщо доступ не підтверджено, сторінка проходження курсу не повинна відкриватися.

Під час створення заявки на оплату система повинна зберігати дані про користувача, курс або вебінар, суму, валюту, статус заявки та посилання для оплати, щоб забезпечити подальшу обробку заявки. Після підтвердження оплати адміністратором користувач повинен отримати доступ до відповідного курсу або вебінару.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

Вимоги до надійності передбачають контроль вхідних даних, захист приватних сторінок, перевірку ролей користувачів і обмеження доступу до адміністративної панелі, щоб зменшити ризик помилок і несанкціонованих дій. Паролі користувачів повинні зберігатися у хешованому вигляді. Для авторизації використовується JWT-токен. Доступ до адміністративних функцій має бути дозволений лише користувачам із роллю admin.

Сайт повинен обробляти помилкові ситуації, зокрема неправильний email або пароль, спробу доступу без авторизації, відсутність курсу, відсутність вебінару, помилку завантаження даних, некоректну заявку або недоступність зовнішнього сервісу.

Умови експлуатації передбачають роботу сайту в сучасних браузерях Chrome, Edge та Firefox. Для коректної роботи потрібне стабільне інтернет-з'єднання, оскільки частина даних завантажується з backend API, MongoDB Atlas, Cloudinary, Monobank payment URL, Google Meet, YouTube та Google Drive. Користувач може працювати із сайтом з персонального комп'ютера, ноутбука, планшета або смартфона.

Сайт не належить до систем реального часу, тому жорсткі часові обмеження не встановлюються. Водночас сторінки повинні відкриватися без значних затримок, а API-запити мають повертати результат у прийнятний для користувача час. Для покращення сприйняття завантаження на клієнтській частині використовуються стани завантаження та тимчасові skeleton-елементи.

Допустиме навантаження на сайт залежить від обраного тарифу хостингу та кількості одночасних користувачів. Для початкової експлуатації сайту психолога достатнім є базовий хмарний хостинг для frontend і backend, оскільки сайт не передбачає складних обчислень у реальному часі.

1.2.4 Вимоги до програмної документації

Програмна документація повинна забезпечувати встановлення, запуск, обслуговування та подальший супровід вебсайту. Вона має бути зрозумілою для

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

розробника, адміністратора та осіб, які будуть підтримувати сайт після завершення розробки.

До складу програмної документації повинні входити: пояснювальна записка до кваліфікаційної роботи, яка визначає мету, завдання та склад проєкту:

- опис структури проєкту;
- опис використаних технологій;
- інструкція з розміщення сайту в Інтернеті;
- інструкція з обслуговування та наповнення сайту;
- інструкція з популяризації та підтримки сайту;
- опис структури бази даних;
- опис основних API-маршрутів;
- інструкція для роботи з адміністративною панеллю;
- матеріали тестування;
- Графічна частина з діаграмами та схемами.

Під час підготовки пояснювальної записки та впорядкування окремих текстових матеріалів використовувалися допоміжні цифрові інструменти для уточнення формулювань, структурування змісту та перевірки логічної послідовності викладу матеріалу [13].

У програмному коді мають використовуватися зрозумілі назви файлів, компонентів, функцій і змінних. Основні модулі мають бути структуровані за призначенням: компоненти інтерфейсу, сторінки, API-клієнти, контексти, маршрути, контролери, сервіси, моделі та middleware.

1.2.5 Техніко-економічні показники

Розробка вебсайту виконувалася із застосуванням сучасних безкоштовних або умовно безкоштовних технологій. Для клієнтської частини використано React, Vite та Tailwind CSS. Для серверної частини використано Node.js, Express.js і MongoDB Atlas. Зображення зберігаються у Cloudinary, а розміщення сайту орієнтоване на Vercel [10] і Render [11].

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

Використання таких технологій дозволяє зменшити витрати на розробку та експлуатацію, оскільки більшість інструментів мають безкоштовні тарифи, достатні для початкового запуску сайту. Це безпосередньо вирішує завдання запуску проєкту з мінімальними початковими витратами. Додаткові витрати можуть виникати у разі підключення власного доменного імені, збільшення обсягу зберігання даних, підвищення навантаження на сервер або переходу на платні тарифи хостингу. Основні техніко-економічні показники проєкту наведено в таблиці 1.2.

Таблиця 1.2 – Техніко-економічні показники вебсайту

Показник	Значення
Тип програмного продукту	Full-stack вебсайт
Frontend-технології	React, Vite, Tailwind CSS
Backend-технології	Node.js, Express.js
База даних	MongoDB Atlas
Сервіс зображень	Cloudinary
Орієнтовний спосіб розміщення frontend	Vercel
Орієнтовний спосіб розміщення backend	Render
Кількість основних ролей	Користувач, адміністратор
Основні розділи	Головна, послуги, курси, вебінари, відгуки, контакти, кабінет, адміністративна панель
Орієнтовні витрати на початковий запуск	Мінімальні, за умови використання безкоштовних тарифів

Економічна доцільність розробки полягає в тому, що сайт об'єднує кілька функцій в одному середовищі: презентацію послуг, керування курсами, реєстрацію на вебінари, особистий кабінет та адміністративне керування. Це безпосередньо розв'язує потребу в окремих сервісах і спрощує підтримку актуальності інформації.

1.2.6 Стадії та етапи розробки

Розробка вебсайту виконувалася поетапно, що дозволило поступово перейти від аналізу предметної області до створення готового програмного продукту, його тестування та підготовки до розміщення в Інтернеті. Основні стадії та етапи розробки наведено в таблиці 1.3.

Таблиця 1.3 – Стадії та етапи розробки вебсайту

Етап	Зміст робіт
1	2
Аналіз предметної області	Вивчення потреб сайту психолога та аналіз існуючих рішень
Формування вимог	Визначення функцій, ролей, сторінок і структури сайту
Проектування структури	Розробка карти сайту, навігації та структури сторінок
Проектування бази даних	Визначення основних колекцій MongoDB та зв'язків між ними
Розробка frontend	Створення сторінок, компонентів, маршрутизації та адаптивної верстки
Розробка backend	Створення API, моделей, контролерів, сервісів і middleware
Розробка адміністративної панелі	Реалізація інтерфейсу для керування контентом

Продовження таблиці 1.3

1	2
Інтеграція сервісів	Підключення MongoDB Atlas, Cloudinary, Monobank payment URL та інших сервісів
Тестування	Перевірка функціоналу, форм, авторизації, навігації та адаптивності
Підготовка до розміщення	Налаштування Vercel, Render, змінних середовища та домену
Оформлення документації	Підготовка пояснювальної записки, схем, таблиць і додатків

На кожному етапі виконувалася перевірка проміжних результатів, що давала змогу виявляти помилки, уточнювати структуру сайту та вдосконалювати функції.

1.2.7 Порядок тестування та прийому

Порядок тестування та приймання вебсайту передбачає перевірку відповідності розробленого продукту технічному завданню. Тестування підтверджує працездатність основних функцій, правильність навігації, авторизації, особистого кабінету, адміністративної панелі та взаємодії з базою даних.

Під час тестування необхідно перевірити:

- відкриття головної сторінки;
- роботу головного меню;
- відкриття сторінок послуг, курсів, вебінарів, відгуків і контактів;
- відкриття детальної сторінки курсу;
- реєстрацію нового користувача;
- вхід користувача в систему;
- вихід із системи;
- роботу особистого кабінету;

- створення заявки на оплату курсу;
- реєстрацію на вебінар;
- доступ до навчальних матеріалів після підтвердження;
- заборону доступу до захищених сторінок без авторизації;
- вхід адміністратора;
- створення, редагування та видалення контенту в адміністративній панелі;
- підтвердження та відхилення заявок на оплату;
- відображення помилкових повідомлень;
- адаптивність сайту на різних екранах;
- робота сайту в браузерях Chrome, Edge та Firefox.

Приймання роботи виконується після успішного проходження основних тестів. Вебсайт вважається готовим до експлуатації, якщо відкриваються всі основні сторінки, навігація працює коректно, користувач може зареєструватися та авторизуватися, адміністративна панель доступна лише адміністратору, дані зберігаються у базі даних, а публічна частина сайту коректно відображається на різних пристроях.

Результати тестування доцільно подати у вигляді checklist у розділі тестування або в додатках до пояснювальної записки. Це наочно підтверджує, що основні функції вебсайту перевірені та працюють відповідно до технічного завдання.

Отже, перший розділ визначає загальне обґрунтування розробки, аналіз наявних рішень і технічне завдання на створення вебсайту для психолога Ольги Демедчук. Сформульовані вимоги є основою для подальшого проектування структури сайту, бази даних, програмної реалізації та тестування, а також для перевірки того, що розробка вирішує поставлені завдання.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури сайту і вебсторінок

Під час розробки вебсайту для психолога Ольги Демедчук було важливо спроектувати його загальну структуру, логіку навігації та взаємозв'язки між основними сторінками. Від правильності побудови структури сайту залежать зручність його використання, швидкість пошуку потрібної інформації та ефективність взаємодії користувача з функціональними можливостями системи.

Розроблений вебсайт призначений для представлення професійної діяльності психолога, інформування відвідувачів про послуги, курси, вебінари, відгуки та способи зв'язку. Крім інформаційної частини, у проєкті реалізовано навчальний функціонал, особистий кабінет користувача та адміністративну панель для керування основним наповненням сайту. Таким чином, сайт охоплює презентаційну та прикладну частини, забезпечуючи взаємодію між користувачем, навчальними матеріалами та адміністратором.

На етапі проєктування було визначено, що структура сайту повинна бути простою для звичайного відвідувача, але водночас достатньо гнучкою для підтримки курсів, вебінарів, особистого кабінету та адміністративного керування. Тому сторінки сайту поділено на публічні та захищені, щоб розмежувати відкритий доступ і функції, доступні після авторизації, та спростити користувачеві навігацію між різними сценаріями взаємодії. Публічні сторінки доступні всім відвідувачам без авторизації. Захищені сторінки відкриваються лише після входу користувача в систему.

Основна маршрутизація сайту реалізована у клієнтській частині за допомогою бібліотеки React Router DOM. У файлі App.jsx описано маршрути, за якими користувач переходить між сторінками, щоб швидко знаходити потрібні розділи та виконувати потрібні дії. Для зручності користування також передбачено перенаправлення з маршруту /home на головну сторінку сайту.

До основних публічних сторінок належать головна сторінка, сторінка

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

послуг, каталог курсів, детальна сторінка курсу, сторінка вебінарів, сторінка відгуків, контактна сторінка, сторінки входу, реєстрації, відновлення та скидання пароля. Вони забезпечують доступ користувача до потрібної інформації та виконання дій. Також реалізовано сторінку помилки 404, яка відкривається у разі переходу на неіснуючий маршрут.

Головна сторінка є центральним елементом сайту. Вона формує перше враження про спеціаліста та містить основні інформаційні блоки. Її структура побудована таким чином, щоб користувач поступово ознайомився з діяльністю психолога, зрозумів напрямки роботи та міг перейти до конкретної дії. Головна сторінка містить такі секції: головний екран, короткий блок про психолога, переваги звернення, попередній перегляд послуг, рекомендовані курси, відгуки, часті запитання та заклик до дії.

Сторінка послуг призначена для подання напрямків роботи психолога. На ній користувач може ознайомитися з доступними послугами, їх коротким описом і перейти до подальшої взаємодії. Каталог курсів містить перелік навчальних продуктів, доступних користувачам. Кожен курс має окрему детальну сторінку, на якій подано опис, програму, вартість і можливість придбання.

Окремо реалізовано сторінку проходження курсу. Вона належить до захищених розділів, оскільки доступ до навчальних матеріалів має лише авторизований користувач, для якого підтверджено доступ до відповідного курсу. Це забезпечує надання матеріалів лише тим користувачам, які мають підтверджений доступ, і розмежовує інформаційну частину сайту та навчальний функціонал.

Сторінка вебінарів призначена для перегляду доступних онлайн-заходів. Користувач може ознайомитися з описом вебінару, датою проведення, тривалістю, цільовою аудиторією та умовами участі. У разі потреби користувач може зареєструватися на вебінар. Якщо вебінар є платним, доступ до посилання на онлайн-зустріч надається після підтвердження оплати адміністратором, щоб користувач отримував доступ до участі лише після виконання необхідної умови.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

Сторінка відгуків використовується для демонстрації думок користувачів або клієнтів про роботу психолога. Контактна сторінка містить інформацію для зв'язку та спрямовує користувача на подальшу комунікацію. Сторінки входу та реєстрації забезпечують створення облікового запису та доступ до персоналізованих функцій сайту.

Захищена частина сайту представлена в особистому кабінеті користувача, щоб забезпечити доступ до персональних функцій після авторизації. До неї входять сторінки огляду, курсів користувача, записів або вебінарів, а також сторінка профілю. У кабінеті користувач може переглядати власні доступи, навчальні матеріали, записи вебінарів і персональні дані, що пов'язує захищений розділ із потребою у персональному доступі.

Основна навігація сайту складається з таких пунктів: «Головна», «Послуги», «Курси», «Вебінари», «Відгуки», «Контакти». Такий набір пунктів є достатнім для швидкого переходу до основних розділів і не перевантажує користувача зайвими елементами. Нижній блок сайту також містить посилання на ключові розділи, що підвищує зручність користування після перегляду сторінки до кінця. Карта сайту наведена у графічній частині кваліфікаційної роботи.

У межах сайту передбачено кілька основних сценаріїв взаємодії користувача із системою: ознайомлення з діяльністю психолога через головну сторінку та сторінку послуг, перегляд курсів і створення заявки на оплату, реєстрація на вебінар, робота з особистим кабінетом після авторизації та керування контентом сайту через адміністративну панель для адміністратора.

У результаті проєктування було сформовано структуру вебсайту, що забезпечує зрозумілий перехід між основними розділами, розподіл функцій між відвідувачем, авторизованим користувачем та адміністратором, а також зручне виконання передбачених дій у межах роботи системи.

2.2 Створення та верстка сторінок сайту

Після визначення структури сайту було створено та виконано верстку його

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

сторінок. Вона здійснювалася з урахуванням вимог до адаптивності, зручності користування, візуальної цілісності та можливості подальшого розширення проекту.

Клієнтську частину вебсайту створено з використанням React та Vite, щоб забезпечити побудову інтерфейсу на основі компонентів і швидке середовище розробки та збирання frontend-частини. Такий підхід дозволяє поділити інтерфейс на окремі логічні блоки, які можна повторно використовувати на різних сторінках.

Для оформлення сторінок використано Tailwind CSS та CSS-змінні, щоб швидко створювати адаптивну верстку і централізовано керувати кольорами, темами та іншими параметрами дизайн-системи. Основні стилі проекту описано у файлі index.css.

Загальний макет сторінок формується за допомогою компонента Layout, щоб об'єднати верхню навігаційну панель, основний вміст сторінки та нижній блок сайту. Компонент Navbar відповідає за головне меню, Footer — за нижню частину сторінки, а основний контент розміщується у центральній області макета. Для плавних переходів і появи окремих блоків використовується бібліотека Framer Motion. Структуру загального макета сторінки наведено на рисунку 2.1.

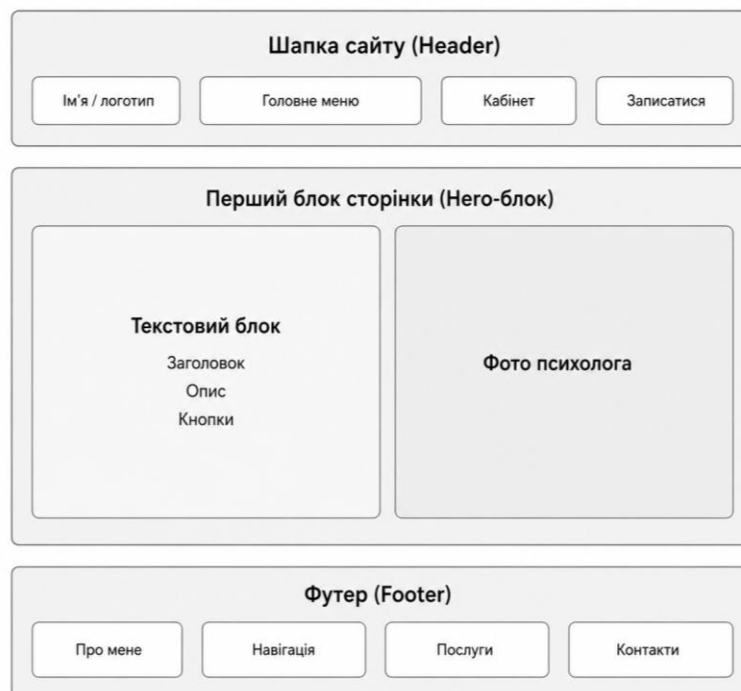


Рисунок 2.1 – Загальна структура макета сторінки

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

Під час верстки використано компонентний підхід. Повторювані елементи інтерфейсу винесено в окремі компоненти: кнопки, картки, поля форм, зображення, блоки завантаження, повідомлення про помилки та порожні стани. Завдяки цьому сторінки мають єдиний стиль, а зміни в окремих елементах можна виконувати без редагування кожної сторінки. Основні компоненти клієнтської частини наведено в таблиці 2.1.

Таблиця 2.1 – Основні компоненти клієнтської частини сайту

Компонент	Призначення
Layout	Формує загальну структуру сторінки
Navbar	Відповідає за верхню навігацію
Footer	Відображає нижній інформаційний блок
PageHero	Формує вступний блок сторінки
Button	Забезпечує єдиний вигляд кнопок
Card	Використовується для карток курсів, послуг і відгуків
FormField	Використовується для полів форм
LazyImage	Оптимізує відображення зображень
LoadingState	Відображає стан завантаження
Skeleton	Показує тимчасовий макет під час завантаження даних
ThemeToggle	Забезпечує перемикання теми оформлення

Дизайн клієнтської частини сайту побудований у темній кольоровій гамі. Основним фоном є темно-зелений колір, який поєднується з глибшими зеленими відтінками, світлим кольором заголовків і золотим акцентом. Така палітра відповідає тематиці сайту психолога, створює спокійний і професійний візуальний образ, а також допомагає виокремити важливі елементи інтерфейсу.

Для заголовків використано шрифти із засічками, зокрема Georgia, Palatino Linotype та Times New Roman. Основний текст оформлено сучасними системними шрифтами, серед яких Segoe UI, system-ui, Roboto та sans-serif. Таке поєднання робить заголовки виразними, а основний текст — зручним для читання.

Сайт має темну тему за замовчуванням. У файлі main.jsx встановлюється значення data-theme="dark", а обрана тема зберігається в localStorage. У стилях також передбачено підтримку світлої теми через селектор html[data-theme="light"]. Це дає змогу в майбутньому розширити можливості для персоналізації зовнішнього вигляду сайту.

Особливу увагу приділено адаптивності. Сайт повинен коректно відображатися на мобільних телефонах, планшетах, ноутбуках і великих моніторах. Для цього використано систему breakpointів Tailwind CSS. Сітки сторінок будуються за допомогою адаптивних класів, зокрема sm, lg, max-w-bxl, rx-4. Це дає змогу автоматично змінювати кількість колонок, відступи та розміщення блоків залежно від ширини екрана.

На мобільних пристроях головна навігація замінюється на burger-меню. Його робота керується станом menuOpen у компоненті Navbar. На великих екранах відображається повна горизонтальна навігація. Такий підхід забезпечує зручність користування клієнтською частиною сайту на різних пристроях.

Для покращення взаємодії з користувачем у клієнтській частині використовуються повідомлення від react-hot-toast. Вони відображають результат виконання дій, наприклад, успішну авторизацію, помилку введення даних або результат надсилання запиту. У випадках, коли дані ще завантажуються з сервера, відображаються loading states або skeleton-елементи. Якщо дані відсутні або виникає помилка, користувач бачить відповідний інформаційний стан, а не порожню сторінку.

Таким чином, створення та верстка клієнтської частини сайту виконані з урахуванням сучасних вимог до вебінтерфейсів: адаптивності, повторного використання компонентів, зрозумілої навігації, візуальної цілісності та зручності

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

користування.

2.3 Розробка структури бази даних сайту

Для зберігання даних вебсайту використовується MongoDB Atlas. Оскільки MongoDB є документно-орієнтованою системою керування базами даних, у ній замість таблиць використовуються колекції.

Структура бази даних описана за допомогою моделей Mongoose, розміщених у папці backend/src/models. Mongoose використовується для визначення схем документів, типів полів, зв'язків між колекціями та правил роботи з даними. Такий підхід дозволяє поєднати гнучкість MongoDB із більш контрольованою структурою даних у backend-частині.

Під час проєктування бази даних було визначено основні об'єкти предметної області: користувачі, послуги, курси, модулі, уроки, доступи до курсів, замовлення, платежі, вебінари, реєстрації на вебінари, відгуки, фотографії та медіафайли. Для кожного об'єкта створено окрему колекцію, щоб окремо підтримувати відповідні дані та їх використання в роботі сайту.

Основні колекції бази даних наведено в таблиці 2.2.

Таблиця 2.2 – Основні колекції бази даних вебсайту

Колекція	Призначення
1	2
users	Зберігання користувачів та адміністраторів
photos	Зберігання посилань на фотографії
services	Зберігання послуг психолога
courses	Зберігання онлайн-курсів
modules	Зберігання модулів курсів
lessons	Зберігання уроків
enrollments	Зберігання доступів користувачів до курсів

Продовження таблиці 2.2

1	2
orders	Зберігання заявок на оплату
payments	Зберігання інформації про платежі
webinars	Зберігання вебінарів
webinar_registrations	Зберігання реєстрацій на вебінари
reviews	Зберігання користувацьких відгуків
testimonials	Зберігання відгуків, якими керує адміністратор

Колекція users використовується для зберігання облікових записів користувачів і адміністраторів. Вона містить електронну пошту, хеш пароля, роль користувача, ім'я, прізвище, телефон, службові поля для відновлення пароля, а також дати створення та оновлення запису. Поле role розмежовує права доступу, а значення user або admin підтримують відповідну логіку роботи сайту.

Колекція photos виконує роль централізованого сховища посилань на зображення. Самі зображення зберігаються у сервісі Cloudinary, а в базі даних зберігається URL-адреса файлу. Такий підхід зменшує навантаження на backend і спрощує роботу із зображеннями під час їх використання на сайті.

Колекція services містить інформацію про послуги психолога. До її основних полів належать назва, опис, посилання, порядок відображення, ознака активності та посилання на фото. Це дає змогу адміністратору керувати послугами та змінювати їхнє відображення на сайті.

Колекція courses використовується для зберігання онлайн-курсів. Вона містить назву курсу, унікальний slug, короткий і повний опис, програму, ціну, тривалість, фотографію, порядок відображення, ознаку публікації та ознаку рекомендованого курсу. Курси пов'язані з модулями, уроками, замовленнями та доступом користувачів, що забезпечує повний цикл їх використання на сайті.

Колекція modules зберігає модулі, що належать до конкретного курсу.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

Кожен модуль містить назву, опис, порядок відображення та посилання на курс. Колекція lessons містить уроки, що входять до модулів. Уроки можуть бути текстовими, відеоуроками або PDF-матеріалами. Для цього передбачено поля contentType, content, videoUrl, pdfUrl, photo, pdf та durationMinutes.

Колекція enrollments призначена для зберігання доступів користувачів до курсів. Вона містить посилання на користувача, курс, замовлення, прогрес проходження, список завершених уроків, дату придбання та статус. Саме ця колекція використовується для перевірки права доступу користувача до навчальних матеріалів і для контролю проходження курсу.

Колекція orders використовується для зберігання заявок на оплату курсів і вебінарів. Вона містить користувача, тип об'єкта, курс або вебінар, суму, валюту, статус, посилання на оплату, коментар адміністратора та дату підтвердження. Зміна статусу замовлення є важливою частиною бізнес-логіки сайту, оскільки визначає подальшу обробку заявки.

Колекція payments містить інформацію про платежі, зокрема про замовлення, суму, статус, провайдера та ідентифікатор транзакції. У поточній реалізації основна логіка підтвердження доступу користувача працює через колекцію orders, а payments виконує допоміжну роль у фіксації платіжних даних.

Колекція webinar містить інформацію про вебінари. Вона містить назву, slug, короткий і повний опис, аудиторію, ознаку безкоштовності, ціну, дату проведення, тривалість, посилання на Google Meet, максимальну кількість учасників, автора, порядок відображення та ознаку активності.

Колекція webinar_registrations використовується для зберігання реєстрацій користувачів на вебінари. Вона містить посилання на користувача, вебінар, замовлення, статус реєстрації, статус оплати, нотатки користувача, службові нотатки адміністратора, дату оплати та дату підтвердження. Для запобігання повторній реєстрації одного користувача на той самий вебінар використовується унікальний індекс user + webinar, що підтримує коректність реєстрацій.

Колекції reviews і testimonials мають схоже призначення, але виконують

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

різні ролі. Reviews використовуються для відгуків, які створюють користувачі. Такі відгуки можуть проходити модерацию. Testimonials — це відгуки, якими керує адміністратор і які використовуються для відображення на публічних сторінках сайту.

Колекція media використовується для зберігання метаданих медіафайлів. Вона містить назву файлу, MIME-тип, розмір, провайдера, URL-адресу, мініатюру, ширину, висоту та дати створення й оновлення. У поточній реалізації пряме завантаження медіафайлів має службовий характер. ER-діаграму бази даних вебсайту подано у графічній частині кваліфікаційної роботи. У таблиці 2.3 наведено основні зв'язки між колекціями.

Таблиця 2.3 – Зв'язки між колекціями бази даних

Зв'язок	Опис
1	2
users – orders	Один користувач може мати багато замовлень
users – enrollments	Один користувач може мати багато доступів до курсів
users – webinar_registrations	Один користувач може мати багато реєстрацій на вебінари
users – reviews	Один користувач може залишати відгуки
photos – services	Одна фотографія може бути пов'язана з послугою
photos – courses	Одна фотографія може бути пов'язана з курсом
photos – testimonials	Одна фотографія може бути пов'язана з відгуком
courses – modules	Один курс складається з багатьох модулів

Продовження таблиці 2.3

1	2
modules – lessons	Один модуль складається з багатьох уроків
courses – enrollments	Один курс може мати багато доступів
modules – lessons	Один модуль складається з багатьох уроків
courses – enrollments	Один курс може мати багато доступів

Розроблена структура бази даних дає змогу зберігати всі необхідні дані для роботи сайту, підтримувати зв'язки між курсами, уроками, користувачами, вебінарами та оплатами, а також забезпечувати роботу адміністративної панелі без прямого редагування бази даних, що підкреслює її відповідність поставленим вимогам.

2.4 Програмування сайту

Програмування сайту виконано за full-stack підходом, який передбачає наявність клієнтської та серверної частин. Клієнтська частина відповідає за відображення інтерфейсу, маршрутизацію, взаємодію користувача зі сторінками та надсилання запитів до сервера. Серверна частина відповідає за обробку запитів, роботу з базою даних, авторизацію, керування курсами, вебінарами, замовленнями, відгуками та адміністративними діями.

Для клієнтської частини використано React, Vite, React Router DOM та Tailwind CSS, а також Framer Motion і react-hot-toast. Для серверної частини використано Node.js, Express.js, MongoDB Atlas та Mongoose, а також JWT, bcryptjs, cookie-parser, cors, helmet, express-rate-limit, dotenv, pino та pino-http.

Backend-частина має багаторівневу структуру. API-маршрути розміщені в папці routes, контролери — в controllers, бізнес-логіка — в services, моделі бази даних — в models, middleware — в middleware, а допоміжні функції — в utils. Така структура дозволяє розділити відповідальність між частинами програми та

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

спростити підтримку коду. Структура backend-частини подана на рисунку 2.5.

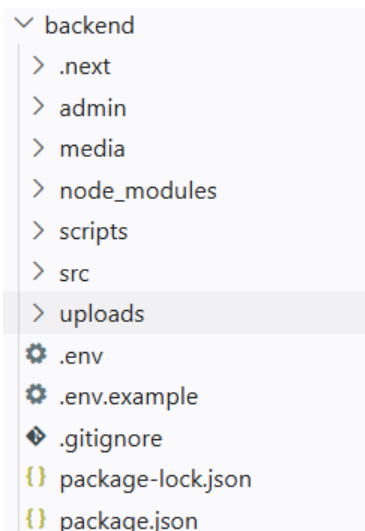


Рисунок 2.2 – Структура backend-частини сайту

Основні API-маршрути наведено в таблиці 2.4.

Таблиця 2.4 – Основні API-маршрути вебсайту

API-маршрут	Призначення
/api/users	Реєстрація, вхід, вихід, профіль,
/api/services	Отримання послуг
/api/courses	Робота з курсами та навчальними матеріалами
/api/progress	Оновлення прогресу проходження уроків
/api/webinars	Отримання вебінарів і реєстрація на них
/api/bookings	Запис на вебінар
/api/orders	Створення та перегляд замовлень
/api/me/dashboard	Дані особистого кабінету
/api/reviews	Робота з відгуками
/api/admin	Адміністративне керування
/api/media	Службові маршрути для медіаданих
/api/health	Перевірка працездатності backend

Авторизацію користувачів реалізовано за допомогою JWT. Після успішного входу або реєстрації користувач отримує токен, який зберігається в localStorage за ключем psytho_auth_token. Під час захищених запитів токен передається в заголовку Authorization. На сервері middleware requireAuth перевіряє наявність і правильність токена, а middleware requireAdmin додатково перевіряє роль користувача для доступу до адміністративних функцій.

Під час реєстрації користувач вводить необхідні облікові дані, які передаються на сервер. Backend перевіряє отриману інформацію, створює користувача, хешує пароль за допомогою bcrypt і повертає токен авторизації. Під час входу система перевіряє email і пароль, після чого оновлює стан авторизації на клієнті.

Сценарій купівлі курсу передбачає відкриття детальної сторінки курсу, натискання кнопки «Придбати» та створення замовлення. Якщо користувач не авторизований, система перенаправляє його на сторінку входу. Якщо користувач авторизований, створюється замовлення та відкривається URL для оплати Monobank. Після підтвердження оплати адміністратором користувач отримує доступ до курсу через запис у enrollments. Послідовність дій користувача під час купівлі курсу та отримання доступу до навчальних матеріалів відображено у графічній частині кваліфікаційної роботи.

Сценарій проходження курсу реалізовано через захищений маршрут /courses//learn/*. Під час відкриття сторінки backend перевіряє доступ користувача до курсу. Якщо доступ підтверджено, система повертає модулі та уроки. Уроки можуть містити текст, відео або PDF-матеріали. Після проходження уроку користувач може позначити його як завершений, а система оновлює прогрес у курсі.

Сценарій реєстрації на вебінар залежить від типу вебінару. Якщо вебінар безкоштовний, система створює підтвержену реєстрацію. Якщо вебінар платний, створюються реєстрація та замовлення на оплату. Посилання Google Meet відображається користувачеві лише після підтвердження реєстрації або оплати.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

Для роботи із зображеннями використовується Cloudinary. Адміністратор завантажує зображення через адміністративну панель, сервер передає його до Cloudinary, отримує URL-адресу та зберігає її у базі даних. Надалі це зображення може використовуватися для курсів, послуг або відгуків.

Надійність роботи серверної частини забезпечується хешуванням паролів, JWT-авторизацією, перевіркою ролей користувачів, налаштуванням CORS, використанням helmet, обмеженням частоти запитів за допомогою express-rate-limit і централізованою обробкою помилок. Це дозволяє зменшити ризик некоректного доступу до даних і забезпечити стабільну роботу системи.

2.4.1 Написання клієнтської частини

Клієнтська частина вебсайту розміщена в папці frontend/src. Точкою входу є файл main.jsx, у якому підключаються BrowserRouter, AuthProvider, ScrollToTop, App, Toaster і глобальні стилі. BrowserRouter забезпечує маршрутизацію, AuthProvider зберігає стан користувача, ScrollToTop повертає сторінку на початок після переходу між маршрутами, а Toaster показує повідомлення користувачу.

Основна маршрутизація описана у файлі App.jsx. У ньому визначено публічні й захищені сторінки, редиректи та сторінку помилки. Для обмеження доступу до приватних сторінок використовується компонент ProtectedRoute. Якщо користувач не авторизований, він не може відкрити особистий кабінет або сторінку проходження курсу.

Стан авторизації зберігається у AuthContext. У ньому містяться дані про користувача, стан завантаження, ознака авторизації, ознака адміністратора, а також функції login, register, logout і refresh. Завдяки цьому сторінки та компоненти отримують інформацію про поточного користувача без дублювання логіки.

Для взаємодії з backend API створено окремі API-модулі. Файл payloadClient.js містить базову логіку HTTP-запитів і автоматично додає токен авторизації. Файл auth.js відповідає за реєстрацію, вхід, вихід, отримання даних користувача, оновлення профілю та відновлення пароля. Файл coursesApi.js

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

використовується для роботи з курсами, навчальними матеріалами та прогресом. Файли `ordersApi.js`, `webinarsApi.js`, `reviewsApi.js`, `dashboardApi.js` і `bookingsApi.js` забезпечують взаємодію з відповідними частинами backend.

Публічна частина сайту включає головну сторінку, сторінку послуг, каталог курсів, сторінку вебінарів, сторінку відгуків, контактну сторінку, сторінки входу та реєстрації. На цих сторінках користувач може ознайомитися з інформацією про психолога, переглянути послуги, курси, вебінари та відгуки, а також перейти до авторизації або запису.

Особистий кабінет доступний лише авторизованим користувачам. Він містить сторінку огляду, сторінку курсів користувача, сторінку записів або вебінарів і сторінку профілю. У цьому розділі користувач може переглядати навчальні матеріали, записи вебінарів і свої дані.

Для покращення користувацького досвіду в інтерфейсі використовуються адаптивне меню, анімації переходів, повідомлення про результати дій, стани завантаження, обробка помилок і повідомлення про відсутність даних. Це допомагає користувачу краще розуміти, що відбувається під час взаємодії із сайтом.

Отже, клієнтська частина забезпечує зручний доступ до основного функціоналу вебсайту, відображення даних із бекенду, авторизацію користувачів, роботу з особистим кабінетом і взаємодію з навчальними матеріалами.

2.4.2 Написання адміністративної частини

Адміністративна частина вебсайту реалізована окремо від основного React-застосунку. Вона розміщена у папці `backend/admin/public` і складається зі статичних HTML-, CSS- та JavaScript-файлів. Такий підхід дозволяє обслуговувати адміністративна панель без окремого frontend-застосунку.

Адміністративна панель відкривається за маршрутом `/admin` і обслуговується сервером Express. Основними файлами адміністративної частини є `index.html`, `admin.js`, `admin-fields.js`, `admin-form.js`, `admin-lookups.js` та `admin.css`.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Файл `index.html` визначає структуру сторінки входу та робочої області адміністратора. Файл `admin.js` містить авторизацію, завантаження даних, CRUD-операції, підтвердження оплат і модерацію. Файл `admin-fields.js` описує поля ресурсів, `admin-form.js` генерує форми, `admin-lookups.js` завантажує довідкові дані, а `admin.css` відповідає за оформлення панелі.

Вхід адміністратора здійснюється через форму з полями `email` і `password`. Дані надсилаються на endpoint `/api/admin/login`. Backend перевіряє облікові дані та роль користувача. Доступ до адміністративної панелі надається лише користувачеві з роллю `admin`.

Адмін-панель дозволяє керувати основними сутностями сайту: користувачами, послугами, курсами, модулями, уроками, вебінарами, фотографіями, відгуками, замовленнями, платежами та доступом. Це дає змогу адміністратору оновлювати наповнення сайту без прямого редагування бази даних.

Особливе значення має керування платежами. Адміністратор може переглядати заявки на оплату, підтверджувати або відхиляти їх. Якщо підтверджується оплата курсу, система створює доступ користувача до курсу. Якщо підтверджується оплата вебінару, відповідна реєстрація отримує статус «Підтверджено». Це дає змогу контролювати доступ користувачів до платних матеріалів.

У розділі доступів адміністративна панель об'єднує два типи даних: доступи до курсів, які зберігаються в колекції `enrollments`, і доступи до вебінарів, які зберігаються в колекції `webinar_registrations`. Найважливіше тут — перегляд усіх записів, пов'язаних із доступом, в одному середовищі, що спрощує адміністрування.

Робота із фотографіями також виконується через адміністративна панель. Адміністратор завантажує зображення, після чого backend передає його в Cloudinary. У MongoDB зберігається лише URL-адреса файлу. Перед видаленням фотографії система перевіряє, чи вона використовується в інших сутностях. Це зменшує ризик випадкового порушення цілісності даних.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

Таким чином, адміністративна частина забезпечує керування контентом, користувачами, курсами, вебінарами, оплатами та відгуками, а також дозволяє підтримувати сайт в актуальному стані без прямого доступу до бази даних.

2.5 Тестування вебсайту

Тестування вебсайту виконувалося для перевірки працездатності основних функцій, правильності маршрутизації, стабільності frontend- та backend-частин, коректності форм, авторизації, особистого кабінету, адміністративної панелі та адаптивності інтерфейсу.

У процесі перевірки було використано функціональне тестування, тестування інтерфейсу, навігації, форм, авторизації, особистого кабінету, адміністративної панелі, API-запитів [12], CRUD-операцій, адаптивне та кросбраузерне тестування, а також базове тестування безпеки.

Функціональне тестування передбачало перевірку основних сценаріїв роботи користувача. Було перевірено відкриття головної сторінки, перехід до сторінки послуг, відкриття каталогу курсів, перегляд детальної сторінки курсу, створення заявки на оплату, авторизацію, відкриття особистого кабінету, перегляд навчальних матеріалів і реєстрацію на вебінар.

Тестування навігації включало перевірку головного меню, посилань у footer, кнопок переходу, сторінки 404 та редиректу з /home на /. Також перевірялося, чи не виникає помилка після перезавантаження вкладених сторінок. Для цього у production-конфігурації frontend використовується файл vercel.json, у якому передбачено rewrite на index.html. Це необхідно для коректної роботи React Router після прямого відкриття сторінок.

Тестування форм охоплювало сторінки входу, реєстрації, відновлення пароля, скидання пароля, оновлення профілю, створення заявок, реєстрації на вебінар та адміністративні форми. Перевірялися правильні та помилкові варіанти введення даних, зокрема порожні поля, неправильний email, неправильний пароль і спроба доступу без авторизації.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

Тестування авторизації передбачало перевірку створення нового користувача, входу, виходу, збереження токена, оновлення стану AuthContext і доступу до захищених маршрутів.

Тестування особистого кабінету включало перевірку сторінок /dashboard, /dashboard/courses, /dashboard/bookings і /dashboard/profile. Було перевірено відображення доступних курсів, записів на вебінари, даних профілю та можливості оновлення персональної інформації.

Тестування адміністративної панелі охоплювало вхід адміністратора, завантаження даних, створення, редагування та видалення записів, модерацію відгуків, підтвердження оплат, керування курсами, модулями, уроками, вебінарами, фотографіями та користувачами. Checklist тестування вебсайту наведено в таблиці 2.5.

Таблиця 2.5 – Checklist тестування вебсайту

Перевірка	Очікуваний результат
1	2
Відкриття головної сторінки	Сторінка завантажується без помилок
Перехід через головне меню	Відкриваються відповідні сторінки
Перехід через footer	Посилання працюють коректно
Відкриття сторінки послуг	Послуги відображаються на сторінці
Відкриття каталогу курсів	Курси завантажуються з backend
Відкриття детальної сторінки курсу	Відображається опис, програма та ціна
Купівля курсу авторизованим користувачем	Створюється замовлення

Продовження таблиці 2.5

1	2
Купівля курсу без авторизації	Користувач переходить на сторінку входу
Відкриття сторінки навчання	Доступ дозволено лише за наявності enrollment
Позначення уроку як завершеного	Прогрес курсу оновлюється
Відкриття сторінки вебінарів	Вебінари відображаються коректно
Реєстрація на вебінар	Створюється реєстрація
Перевірка Google Meet Link	Посилання доступне лише після підтвердження
Вхід користувача	Користувач отримує доступ до кабінету
Реєстрація користувача	Створюється новий обліковий запис
Відновлення пароля	Запит обробляється системою
Відкриття особистого кабінету	Дані користувача відображаються
Оновлення профілю	Зміни зберігаються
Відкриття /admin	Відображається форма входу адміністратора
Вхід адміністратора	Відкривається адміністративна панель
CRUD курсів	Курси створюються, редагуються та видаляються

Продовження таблиці 2.5

1	2
CRUD модулів	Модулі коректно прив'язуються до курсів
CRUD уроків	Уроки створюються та відображаються
Завантаження фото	Фото завантажуються у Cloudinary
Підтвердження оплати	Користувач отримує доступ
Відхилення оплати	Статус заявки змінюється
Модерація відгуків	Відгук отримує відповідний статус
Перевірка ширини 320px	Інтерфейс коректний на мобільному екрані
Перевірка ширини 768px	Інтерфейс коректний на планшеті
Перевірка ширини 1024px	Інтерфейс коректний на ноутбучі
Перевірка ширини 1920px	Інтерфейс коректний на великому екрані
Перевірка у Chrome	Критичні помилки відсутні
Відкриття неіснуючого маршруту	Відображається сторінка 404

Адаптивне тестування проводилося для різних розмірів екрана: 320px, 768px, 1024px і 1920px. Перевірялися розміщення блоків, робота мобільного меню, читабельність тексту, коректність відображення карток, кнопок, форм і таблиць.

Кросбраузерне тестування виконувалося у сучасних браузерях Chrome, Edge і Firefox. У результаті перевірки встановлено, що основні сторінки, маршрути,

форми та елементи інтерфейсу працюють коректно.

Базове тестування безпеки включало перевірку захисту приватних сторінок, доступу до адміністративної панелі, ролей користувачів, JWT-авторизації, хешування паролів, CORS, Helmet і rate limit. Також перевірялося, що користувач без підтверженого доступу не може переглядати навчальні матеріали курсу.

Надійність роботи сайту забезпечується насамперед JWT-авторизацією, хешуванням паролів, перевіркою ролей користувачів і захистом адміністративних маршрутів, а також CORS-конфігурацією, використанням helmet, обмеженням частоти запитів, централізованою обробкою помилок, перевіркою важливих полів на backend, перевіркою посилань на фотографії, обмеженням доступу до навчання без enrollment та контролем статусів оплат.

Жорсткі часові характеристики для цього вебсайту не є критичними, оскільки система не працює в режимі реального часу. Водночас сторінки повинні завантажуватися без значних затримок, а API-запити мають повертати результат у прийнятний для користувача час. Для покращення сприйняття завантаження використовуються loading states і skeleton-елементи.

Умовами експлуатації сайту є використання сучасного браузера, стабільного інтернет-з'єднання та доступності зовнішніх сервісів, зокрема MongoDB Atlas, Cloudinary, Monobank payment URL, Google Meet, YouTube та Google Drive. Frontend-частина орієнтована на розгортання у Vercel, backend-частина — у Render. Для налаштування production-середовища використовуються файли vercel.json, render.yaml та змінні середовища backend/.env.

Отже, проведене тестування підтвердило працездатність основних функцій вебсайту, коректність навігаційної структури, роботу авторизації, особистого кабінету, адміністративної панелі, API-запитів та адаптивного інтерфейсу. Розроблений вебсайт відповідає поставленій задачі та може використовуватися для представлення діяльності психолога, керування послугами, курсами, вебінарами та взаємодії з користувачами.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Розміщення сайту в Інтернеті

Після завершення основних етапів розробки вебсайту важливим завданням є його розміщення в мережі Інтернет. Розміщення сайту забезпечує доступ користувачів до розробленого програмного продукту з різних пристроїв і браузерів, а також дає змогу перевірити працездатність системи в умовах, наближених до реальної експлуатації.

Розроблений вебсайт для психолога Ольги Демедчук є full-stack вебзастосунком, що складається з клієнтської та серверної частин. Клієнтська частина створена з використанням React, Vite, React Router DOM і Tailwind CSS. Серверна частина реалізована на основі Node.js та Express.js. Для зберігання даних використовується MongoDB Atlas, а для роботи із зображеннями — сервіс Cloudinary. Така архітектура передбачає окреме розгортання frontend- і backend-частин, а також підключення зовнішньої хмарної бази даних.

У процесі підготовки сайту до розміщення в Інтернеті було враховано особливості його структури. Frontend-частина відповідає за відображення сторінок, маршрутизацію, адаптивний інтерфейс, форми, особистий кабінет користувача та взаємодію з API. Backend-частина забезпечує авторизацію, обробку запитів, роботу з базою даних, керування курсами, вебінарами, заявками, відгуками, користувачами та адміністративною панеллю.

Для розміщення frontend-частини доцільно використовувати платформу Vercel. Цей сервіс призначений для розгортання сучасних вебзастосунків і добре підходить для проєктів, створених за допомогою React та Vite. Перед розгортанням frontend-частини необхідно переконатися, що проєкт успішно збирається локально. Для цього в середовищі розробки виконується команда встановлення залежностей, після чого запускається команда збирання проєкту. Якщо під час збирання не виникає помилок, frontend-частину можна передавати на хостинг.

Особливістю розробленого сайту є використання клієнтської маршрутизації

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

React Router DOM. У таких застосунках після перезавантаження сторінки або прямого переходу за вкладеним маршрутом сервер має повертати файл index.html. Для цього у frontend-частині передбачено файл vercel.json, у якому налаштовано перенаправлення всіх маршрутів на index.html. Завдяки цьому сторінки /services, /courses, /webinars, /dashboard та інші маршрути не викликають помилку 404 після перезавантаження.

Загальна процедура розміщення frontend-частини на Vercel складається з таких етапів. Спочатку проєкт завантажують у репозиторій GitHub. Після цього в особистому кабінеті Vercel створюється новий проєкт і підключається відповідний репозиторій. У налаштуваннях проєкту вибирається папка frontend, якщо frontend-частина розміщена в окремому каталозі. Далі задаються команда встановлення залежностей і команда збирання. Для застосунку на Vite результат збирання зазвичай розміщується в папці dist. Після цього вказуються середовищні змінні, необхідні для підключення frontend до API backend.

Для коректної роботи клієнтської частини необхідно вказати адресу backend-сервера. Така адреса використовується API-клієнтами, які надсилають запити до серверної частини. У production-середовищі вона повинна містити повну URL-адресу backend, розміщеного на хостингу. Після завершення налаштувань Vercel виконує автоматичне збирання та публікацію frontend-частини. У результаті сайт отримує технічну адресу у домені Vercel, яку можна використовувати для попередньої перевірки роботи.

Backend-частина сайту орієнтована на розгортання на платформі Render. Цей сервіс дає змогу запускати Node.js-застосунки, підключати змінні середовища та автоматично перезапускати сервер після оновлення коду. Перед розміщенням backend-частини необхідно переконатися, що сервер коректно запускається локально, підключається до MongoDB Atlas і відповідає на запити.

Для розгортання backend-частини на Render створюється новий веб-сервіс. Як джерело коду підключається до репозиторію GitHub? У налаштуваннях сервісу вказуються папка backend, команда встановлення залежностей, команда запуску

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

сервера та версія середовища Node.js. Для автоматизації production-конфігурації в проєкті використовується файл `render.yaml`. Він описує параметри запуску backend-сервісу та спрощує повторне розгортання застосунку.

Важливим етапом розміщення backend-частини є налаштування змінних середовища. До основних змінних належать `MONGO_URI`, `JWT_SECRET`, `FRONTEND_ORIGIN`, `FRONTEND_ORIGINS`, `COOKIE_SECURE`, `TRUST_PROXY`, `MONOBANK_PAYMENT_URL`, `CLOUDINARY_URL`, `CLOUDINARY_CLOUD_NAME`, `CLOUDINARY_API_KEY` та `CLOUDINARY_API_SECRET`. Змінна `MONGO_URI` використовується для підключення до бази даних MongoDB Atlas. `JWT_SECRET` потрібна для створення та перевірки токенів авторизації. `FRONTEND_ORIGIN` і `FRONTEND_ORIGINS` використовуються для правильного налаштування CORS і дозволяють серверу приймати запити від розгорнутої frontend-частини. Змінні Cloudinary забезпечують можливість завантаження та зберігання зображень. `MONOBANK_PAYMENT_URL` використовується для переходу користувача на сторінку оплати курсів або вебінарів.

База даних MongoDB Atlas розміщується окремо від backend-сервера. Для її налаштування створюється кластер, після чого вказуються користувач бази даних, пароль, дозволені IP-адреси та назва бази даних. Для production-середовища важливо обмежити доступ до бази даних і використовувати складні паролі. Після створення кластера MongoDB Atlas надає рядок підключення, який записують у змінну `MONGO_URI` на Render.

Після розгортання frontend- і backend-частин виконується перевірка їхньої взаємодії. Спочатку перевіряється доступність головної сторінки сайту. Далі перевіряється відкриття сторінок послуг, курсів, вебінарів, відгуків, контактів, а також входу та реєстрації. Окремо перевіряється робота API-запитів: отримання курсів, послуг, вебінарів, авторизація користувача, створення замовлення та відкриття особистого кабінету. Також необхідно перевірити адміністративну панель за маршрутом `/admin`.

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

Після успішної перевірки технічних адрес Vercel і Render можна підключати власне доменне ім'я. Для цього реєструється домен у реєстраторі доменів, після чого в його DNS-налаштуваннях додаються записи, які вказують на Vercel. Зазвичай для головного домену використовується А-запис або ALIAS/ANAME-запис, а для піддомену www — CNAME-запис. Якщо backend також має використовувати окремий піддомен, наприклад, api.domain.ua, для нього створюється CNAME-запис, який вказує на адресу Render.

Після зміни DNS-записів необхідно дочекатися їх оновлення. Цей процес може займати від кількох хвилин до кількох годин. Після оновлення DNS перевіряється доступ до сайту за новим доменним ім'ям. Перевірка включає відкриття головної сторінки, переходи між розділами, перезавантаження вкладених сторінок, роботу форм, авторизацію, доступ до особистого кабінету, роботу адміністративної панелі та коректність API-запитів.

Під час перевірки сайту за доменним ім'ям необхідно також переконатися, що використовується захищене HTTPS-з'єднання. Vercel і Render підтримують автоматичне підключення SSL-сертифікатів, тому після правильного налаштування домену сайт повинен відкриватися за адресою, що починається з https://. Використання HTTPS є обов'язковим для безпечної передачі даних авторизації, токенів і запитів користувача.

Окрему увагу потрібно приділити налаштуванню CORS після підключення домену. У змінних середовища backend необхідно вказати фактичну адресу frontend-частини. Якщо цього не зробити, браузер може заблокувати запити до API. Тому після підключення домену змінні FRONTEND_ORIGIN і FRONTEND_ORIGINS оновлюються відповідно до фактичної адреси сайту.

Після завершення розміщення сайту в Інтернеті виконується контрольна перевірка. Перевіряється доступність сторінок, працездатність маршрутизації, завантаження даних із MongoDB Atlas, відображення зображень із Cloudinary, перехід за Monobank payment URL, доступ до навчальних матеріалів, відкриття Google Meet Link для підтверджених вебінарів, а також відсутність критичних

					<i>2026.KBP.122.421.15.00.00 ПЗ</i>	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

помилки у консолі браузера.

Таким чином, розміщення вебсайту в Інтернеті передбачає окреме розгортання frontend-частини на Vercel, backend-частини на Render, підключення до MongoDB Atlas, налаштування зовнішніх сервісів, додавання змінних середовища, підключення доменного імені та перевірку роботи сайту за новою адресою. Виконання цих дій забезпечує повноцінну експлуатацію розробленого сайту користувачами через Інтернет.

3.2 Інструкція з обслуговування та наповнення сайту

Обслуговування та наповнення вебсайту є важливими складовими його експлуатації. Після завершення розробки сайт потребує регулярного оновлення контенту, перевірки працездатності та додавання нових послуг, курсів, вебінарів, відгуків і зображень. Для цього у проєкті реалізовано адміністративну панель, яка дозволяє керувати основними даними без прямого редагування бази даних.

Адміністративна частина сайту реалізована окремо від основного React-застосунку. Вона розміщена у папці backend/admin/public і складається зі статичних HTML-, CSS- та JavaScript-файлів. Admin-панель обслуговується backend-сервером Express за маршрутом /admin. Такий підхід дозволяє адміністратору працювати з даними сайту в браузері без сторонніх засобів керування базою даних.

Для входу в адміністративну панель адміністратор відкриває сторінку /admin. На сторінці відображається форма авторизації, у яку потрібно ввести email і пароль. Після відправлення форми дані передаються на endpoint /api/admin/login. Сервер перевіряє облікові дані користувача та його роль. Доступ до адміністративної частини дозволяється лише користувачам із роллю admin. Якщо користувач не має відповідних прав, доступ до панелі не надається.

Після успішного входу адміністратор переходить до робочої області. У ній доступні основні розділи для керування сайтом: користувачами, послугами, курсами, модулями, уроками, вебінарами, фотографіями, відгуками, замовленнями, платежами та доступом. Кожен розділ має табличне або спискове представлення

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

даних і набір дій для створення, редагування, перегляду або видалення записів.

Загальна послідовність роботи адміністратора із сайтом складається з таких етапів: вхід до адміністративної панелі, вибір потрібного розділу, перегляд наявних записів, створення нового запису або редагування існуючого, збереження змін, перевірка результату в публічній частині сайту. Такий порядок дозволяє уникнути випадкових помилок і контролювати відображення змін для користувачів.

Для редагування послуг адміністратор відкриває відповідний розділ адміністративної панелі. У ньому можна змінити назву послуги, опис, посилання, порядок відображення, активність і пов'язане зображення. Поле активності дозволяє тимчасово приховати послугу з публічної частини сайту без її видалення з бази даних. Це зручно, коли певний напрям роботи тимчасово не використовується, але може бути відновлений у майбутньому.

Для додавання нової послуги адміністратор створює новий запис, заповнює обов'язкові поля, вибирає фотографію та зберігає внесені зміни. Після цього послуга може відобразитися на сторінці послуг або в секції попереднього перегляду на головній сторінці. Перед публікацією бажано перевірити правильність тексту, коректність зображення та зручність відображення на мобільному пристрої.

Керування курсами є одним із найважливіших елементів адміністративної частини. Курс містить назву, slug, короткий опис, повний опис, програму, ціну, тривалість, фотографію, порядок відображення, ознаку публікації та ознаку рекомендованого курсу. Поле isPublished визначає, чи буде курс доступним на сайті. Поле isFeatured дозволяє відображати курс у блоці рекомендованих курсів на головній сторінці.

Під час створення курсу адміністратор спочатку додає основну інформацію про нього. Після цього створюються модулі, що належать до курсу. Кожен модуль має назву, опис і порядок відображення. Далі до модулів додаються уроки. Урок може бути текстовим, відеоуроком або PDF-матеріалом. Для відеоуроків використовується URL YouTube, для PDF-матеріалів — URL PDF у Google Drive

					<i>2026.KBP.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

або пов'язаний media-запис. Така структура дозволяє організувати курс як послідовну навчальну програму.

Редагування курсу виконується у разі зміни опису, вартості, програми, тривалості або зображення. Якщо курс уже має користувачів із доступом, редагування потрібно виконувати обережно, щоб не порушити логіку навчання. Наприклад, зміна порядку модулів або видалення уроків може вплинути на прогрес користувачів. Тому перед суттєвими змінами рекомендується перевірити, чи є активні доступи до курсу.

Окремий розділ адміністративної панелі використовується для керування вебінарами. Адміністратор може створювати нові вебінари, вказувати назву, опис, цільову аудиторію, дату проведення, тривалість, вартість, максимальну кількість учасників і посилання на Google Meet. Якщо вебінар безкоштовний, користувач може отримати підтверджену реєстрацію безкоштовно. Якщо вебінар платний, доступ до посилання на зустріч надається після підтвердження оплати адміністратором.

Під час створення вебінару особливо важливо правильно вказати дату, тривалість і посилання на онлайн-зустріч. Також слід перевірити, чи відповідає опис вебінару його цільовій аудиторії. У проєкті передбачено поділ аудиторії на спеціалістів і клієнтів, що дозволяє точніше структурувати інформацію для різних груп користувачів.

Керування замовленнями та оплатами здійснюється через адміністративну панель. Коли користувач створює заявку на оплату курсу або вебінару, у базі даних створюється запис у колекції orders. Спочатку така заявка має статус pending. Адміністратор переглядає заявку, перевіряє інформацію про оплату та може підтвердити або відхилити її. Якщо оплата курсу підтверджена, система створює доступ користувача до курсу. Якщо оплата вебінару підтверджена, реєстрація користувача отримує статус «підтверджена».

Такий механізм дозволяє адміністратору контролювати доступ до платного контенту. Це особливо важливо для навчальних матеріалів, оскільки сторінка

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

проходження курсу має бути доступна лише користувачам із підтвердженим доступом. Завдяки цьому сайт виконує не лише презентаційну, а й навчальну функцію.

Розділ фотографій використовується для завантаження та керування зображеннями. Адміністратор додає фото через адміністративну панель, після чого backend передає його в Cloudinary. У базі даних зберігається URL-адреса зображення. Надалі це фото можна прив'язати до послуги, курсу або відгуку. Перед видаленням фотографії система перевіряє, чи вона використовується в інших сутностях. Це допомагає запобігти ситуації, коли після видалення зображення на публічній сторінці зникає ілюстрація.

Керування відгуками поділено на два напрями. Користувацькі відгуки зберігаються у колекції reviews і можуть потребувати модерації. Адміністратор перевіряє текст відгуку, його відповідність тематиці сайту та ухвалює рішення щодо публікації. Відгуки, які керуються безпосередньо адміністратором, зберігаються у колекції testimonials. Вони можуть використовуватися для відображення на головній сторінці або на сторінці відгуків.

Для обслуговування користувачів адміністратор може переглядати облікові записи, ролі та пов'язані з ними дії. Важливо, щоб роль admin надавалася лише відповідальним особам, оскільки вона відкриває доступ до керування контентом, платежами та користувацькими правами доступу.

Під час наповнення сайту необхідно дотримуватися єдиного стилю. Назви послуг, курсів і вебінарів мають бути короткими та зрозумілими. Описи повинні бути інформативними, але не перевантаженими. Для курсів бажано вказувати чітку структуру програми, очікуваний результат і тривалість. Для вебінарів потрібно зазначати дату, тему, аудиторію та формат проведення. Для зображень варто використовувати якісні фотографії в одному стилі, щоб не порушувати візуальну цілісність сайту.

Після внесення змін в адміністративній панелі адміністратор повинен перевірити результат на публічній частині сайту. Наприклад, після створення курсу

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

потрібно відкрити каталог курсів і сторінку з детальною інформацією про нього. Після додавання вебінару потрібно перевірити сторінку вебінарів. Після зміни послуги потрібно переглянути сторінку послуг і головну сторінку, якщо послуга відображається у попередньому блоці.

Обслуговування сайту також включає періодичну перевірку працездатності зовнішніх сервісів. До таких сервісів належать MongoDB Atlas, Cloudinary, Monobank payment URL, Google Meet, YouTube та Google Drive. Якщо один із цих сервісів недоступний або посилання вказано неправильно, окремі функції сайту можуть працювати некоректно. Тому адміністратору потрібно періодично перевіряти доступність зображень, навчальних відео, PDF-матеріалів і посилань на вебінари.

Мінімальна конфігурація сервера для хостингу сайту залежить від кількості користувачів і рівня навантаження. Для початкової експлуатації достатньо хмарного Node.js-середовища з підтримкою сучасної версії Node.js, доступом до змінних середовища, HTTPS, автоматичним перезапуском сервісу та можливістю підключення до MongoDB Atlas. Сервер повинен мати достатній обсяг оперативної пам'яті для стабільної роботи Express-застосунку, обробки API-запитів і взаємодії з базою даних. Для невеликого сайту психолога на початковому етапі достатньо базового тарифу Render або аналогічного хостингу для Node.js-застосунків.

Для frontend-частини мінімальні вимоги до хостингу включають підтримку статичних файлів, HTTPS, налаштування перенаправлень для React Router і можливість додавання змінних середовища. Vercel відповідає цим вимогам і дозволяє автоматично оновлювати сайт після внесення змін у репозиторій.

Для стабільної роботи сайту потрібно регулярно виконувати такі дії: перевіряти доступність сайту, переглядати логи backend-сервера, контролювати правильність змінних середовища, оновлювати залежності після перевірки сумісності, перевіряти роботу авторизації, тестувати основні форми, контролювати доступність бази даних і зовнішніх сервісів. Також важливо не зберігати секретні ключі в коді проєкту, а використовувати для цього змінні середовища.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

Таким чином, інструкція з обслуговування та наповнення сайту передбачає роботу з адміністративною панеллю, редагування наявного контенту, створення нових послуг, курсів, модулів, уроків і вебінарів, модерацію відгуків, підтвердження оплат, керування фотографіями та регулярну перевірку працездатності сайту. Реалізована адміністративна частина дозволяє підтримувати сайт в актуальному стані без прямого доступу до бази даних.

3.3 Інструкція з популяризації та підтримки сайту

Після розміщення вебсайту в Інтернеті важливо зайнятися його популяризацією та подальшою підтримкою. Навіть якісно розроблений сайт не буде ефективним, якщо користувачі не зможуть легко знайти його в пошукових системах або соціальних мережах. Тому просування сайту є невід’ємною частиною його експлуатації.

Вебсайт психолога Ольги Демедчук має інформаційний, презентаційний і навчальний характер. Його цільовою аудиторією є люди, які шукають психологічну допомогу, клієнти, зацікавлені в консультаціях, користувачі, які хочуть проходити онлайн-курси, а також учасники вебінарів. З огляду на це, популяризація сайту має бути спрямована на підвищення його видимості в мережі Інтернет, формування довіри до спеціаліста та збільшення кількості цільових звернень.

Одним із основних механізмів популяризації є пошукова оптимізація, тобто SEO. Вона передбачає підготовку сторінок сайту до коректного індексування пошуковими системами. Для цього кожна важлива сторінка повинна мати зрозумілий заголовок, опис, логічну структуру тексту, коректні URL-адреси та зміст, що відповідає запитам цільової аудиторії.

Для головної сторінки важливо використовувати ключові фрази, пов’язані з діяльністю психолога, консультаціями, психотерапією, курсами та вебінарами. Однак такі фрази мають бути вписані природно, без надмірного повторення. Текст має залишатися зрозумілим для реального відвідувача, оскільки основна мета сайту — не лише залучити користувача з пошуку, а й утримати його увагу та спрямувати

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

до конкретної дії.

Сторінка послуг повинна містити чіткий опис напрямів роботи психолога. Для кожної послуги бажано використовувати окремий змістовний опис, у якому пояснюється, кому вона підходить, які проблеми вона допомагає вирішити та як користувач може звернутися. Це підвищує інформативність сторінки та покращує її сприйняття пошуковими системами.

Сторінки курсів також мають важливе значення для популяризації сайту. Кожен курс повинен мати унікальну назву, короткий опис, повний опис, програму, вартість і зрозумілий slug. Чим повніше описано курс, тим легше користувачеві прийняти рішення про його придбання. Крім того, змістовні сторінки курсів можуть залучати користувачів із пошукових систем за тематичними запитами.

Окрему увагу варто приділити сторінці вебінарів. Для кожного вебінару бажано вказувати тему, дату, тривалість, аудиторію, опис і умови участі. Якщо вебінари проводяться регулярно, сторінка може стати важливим джерелом залучення нових користувачів. Після завершення вебінару можна оновлювати інформацію або створювати новий вебінар на схожу тему.

Для покращення пошукової оптимізації потрібно насамперед використовувати метадані сторінок. У проєкті передбачено окремі файли з інформацією про сторінки, зокрема pageMeta.js. Це дає змогу централізовано керувати назвами сторінок та їхніми описами. Коректні title і description допомагають пошуковим системам краще розуміти зміст сторінок і формувати привабливі результати пошуку.

Важливим елементом популяризації є створення зрозумілої навігаційної структури. Сайт має головне меню з розділами «Головна», «Послуги», «Курси», «Вебінари», «Відгуки» та «Контакти». Така структура є зручною для користувача й водночас допомагає пошуковим системам індексувати сторінки. Внутрішні посилання між головною сторінкою, послугами, курсами та вебінарами підвищують зв'язаність сайту.

Для популяризації сайту доцільно використовувати соціальні мережі.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

Оскільки сайт присвячений діяльності психолога, найбільш доцільними каналами можуть бути Instagram, Facebook, YouTube, Telegram або інші платформи, де цільова аудиторія шукає поради, анонси вебінарів і навчальні матеріали. У соціальних мережах можна публікувати короткі інформаційні дописи, анонси курсів, дати вебінарів, фрагменти корисних матеріалів і посилання на відповідні сторінки сайту.

Соціальні мережі також можуть використовуватися для формування довіри до спеціаліста. Для цього важливо регулярно публікувати змістовний контент: відповіді на поширені запитання, пояснення психологічних тем, рекомендації щодо самоспостереження, анонси консультацій, відгуки клієнтів і посилання на матеріали на сайті. При цьому сайт має виконувати роль центрального ресурсу, на який ведуть усі зовнішні канали.

Ще одним способом популяризації є використання контент-маркетингу. Для цього на сайті або в пов'язаних каналах можна регулярно публікувати корисні матеріали на теми психології, психотерапії, емоційного стану, внутрішніх конфліктів, стосунків, самооцінки та особистісного розвитку. Якщо в майбутньому буде додано блог або розділ статей, це може значно покращити органічне просування сайту в пошукових системах.

Для підвищення довіри користувачів важливо підтримувати розділ відгуків. Відгуки допомагають новим відвідувачам краще оцінити досвід інших клієнтів і ухвалити рішення щодо звернення. Оскільки у проєкті реалізовано розділи reviews і testimonials, адміністратор може модерувати користувацькі відгуки та керувати відгуками, які відображаються на публічних сторінках. Водночас потрібно стежити, щоб опубліковані відгуки були коректними, етичними та не містили конфіденційної інформації.

Популяризація сайту також пов'язана з технічною якістю його роботи. Якщо сторінки завантажуються повільно, зображення не завантажуються або форми працюють некоректно, користувачі можуть залишити сайт. Тому підтримка сайту повинна включати регулярну перевірку швидкодії, доступності сторінок,

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

коректності форм, роботи авторизації, завантаження зображень і доступності зовнішніх сервісів.

Для аналізу ефективності просування доцільно насамперед залучити інструменти веб-аналітики. Наприклад, можна використовувати Google Analytics або інші подібні сервіси. Вони дають змогу відстежувати кількість відвідувачів, джерела переходів, популярні сторінки, поведінку користувачів і конверсії. На основі цих даних можна визначити, які сторінки потребують покращення, які курси або вебінари викликають найбільший інтерес і з яких каналів приходять користувачі.

Також доцільно насамперед підключити Google Search Console. Цей інструмент дозволяє перевіряти індексацію сторінок, бачити пошукові запити, за якими користувачі знаходять сайт, виявляти технічні помилки, проблеми з мобільною версією та інші фактори, що впливають на видимість сайту в пошуку. Після підключення домену до Search Console можна контролювати, як сайт індексується пошуковою системою.

Для стабільної підтримки сайту потрібно періодично оновлювати його контент. Якщо на сайті інформація довго не оновлюється, користувач може сприймати його як неактуальний. Тому рекомендується регулярно оновлювати сторінку вебінарів, додавати нові курси, актуалізувати описи послуг, перевіряти контактну інформацію, оновлювати відгуки та за потреби змінювати блоки на головній сторінці.

Окремим напрямом підтримки є контроль безпеки. Оскільки сайт має авторизацію, особистий кабінет, адміністративна панель і платні матеріали, необхідно стежити за надійністю паролів, доступом до змінних середовищ, захистом адміністративного облікового запису та актуальністю залежностей. Секретні ключі, такі як JWT_SECRET, MONGO_URI і ключі Cloudinary, не повинні зберігатися у відкритому коді. Вони мають бути розміщені лише у змінних середовищах хостингу.

					<i>2026.KBP.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

Для підтримки сайту в робочому стані необхідно періодично перевіряти логи backend-сервера. Логи допомагають виявляти помилки в API-запитах, проблеми з підключенням до бази даних, некоректні запити користувачів або збої в роботі зовнішніх сервісів. Регулярний перегляд логів дозволяє своєчасно знаходити причини нестабільної роботи окремих модулів сайту та швидше реагувати на можливі технічні проблеми. Якщо виникають помилки, їх потрібно аналізувати та усувати до того, як вони суттєво вплинуть на користувачів або призведуть до порушення роботи основних функцій вебсайту.

Важливим елементом підтримки є резервне збереження даних. Оскільки сайт використовує MongoDB Atlas, потрібно періодично контролювати стан бази даних і, за можливості, використовувати вбудовані механізми резервного копіювання. Це дозволить зменшити ризик втрати даних про користувачів, курси, вебінари, замовлення та відгуки. Крім того, резервне копіювання є важливим для відновлення роботи сайту у разі випадкового видалення інформації, помилкових змін або технічних збоїв.

Підтримка зображень також має значення. Оскільки фотографії зберігаються у Cloudinary, потрібно стежити за коректністю URL-адрес, якістю зображень і тим, щоб не видаляти фото, які використовуються на сайті. У проекті передбачено перевірку використання фотографій перед видаленням, що допомагає зберігати цілісність контенту. Також доцільно періодично перевіряти, чи всі зображення коректно відображаються на сторінках сайту та не сповільнюють завантаження інтерфейсу.

Для підвищення популярності сайту можна використовувати партнерські згадки. Наприклад, посилання на сайт можуть розміщуватися у профілях психолога, на сторінках професійних спільнот, у соціальних мережах, на сторінках вебінарів, у матеріалах курсів або в електронних листах. Чим більше якісних і релевантних джерел посилаються на сайт, тим вищою може бути його довіра для користувачів і пошукових систем. Такі згадки також допомагають залучати нову аудиторію та спрямовувати користувачів до основних розділів вебсайту.

					<i>2026.КВР.122.421.15.00.00 ПЗ</i>	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

Під час популяризації потрібно враховувати етичність теми. Оскільки сайт пов'язаний із психологічними послугами, просування має бути коректним, виваженим і відповідальним, не містити маніпулятивних обіцянок і не створювати нереалістичних очікувань у користувачів. Тексти мають підкреслювати професійність, довіру, конфіденційність і зрозумілість послуг, а також відповідати загальній тематиці ресурсу. Це особливо важливо, оскільки користувачі можуть звертатися до сайту в пошуку підтримки, інформації або навчальних матеріалів, тому комунікація має бути обережною, зрозумілою та відповідальною.

Для подальшого розвитку сайту можна передбачити такі напрями: додавання блогу, створення сторінок для окремих послуг, покращення SEO-метаданих, додавання sitemap.xml, налаштування robots.txt, підключення веб-аналітики, розширення функцій особистого кабінету, автоматизація підтвердження оплат і створення email-сповіщень для користувачів. Такі вдосконалення можуть поступово підвищити зручність користування сайтом, спростити роботу адміністратора та посилити присутність ресурсу в Інтернеті. Крім того, подальше розширення функціоналу дозволить зробити вебсайт більш гнучким, зручним для супроводу та корисним для його цільової аудиторії.

Отже, популяризація та підтримка вебсайту передбачають комплекс заходів: пошукову оптимізацію, оновлення контенту, роботу із соціальними мережами, підтримку відгуків, контроль технічного стану, аналіз відвідуваності, перевірку безпеки, резервне збереження даних і регулярне вдосконалення функціоналу. Виконання цих дій дозволить підтримувати сайт актуальним, стабільним, зручним для користувачів і помітним у мережі Інтернет. Крім того, системна підтримка забезпечує можливість подальшого розвитку вебсайту відповідно до потреб психолога та його аудиторії, а також допомагає зберігати якість роботи ресурсу протягом тривалого часу.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини кваліфікаційної роботи є виконання розрахунків, що дозволяють оцінити економічну доцільність створення вебсайту для психолога Ольги Демедчук. Економічний розділ дає можливість визначити орієнтовні витрати на розробку програмного продукту, розрахувати собівартість робіт, сформувавши ціну розробки, а також оцінити ефективність використання такого веб-ресурсу після його впровадження.

Об'єктом розробки є вебсайт для психолога Ольги Демедчук, призначений для представлення професійної діяльності спеціаліста, розміщення інформації про послуги, курси, вебінари, відгуки та контактні дані. Крім інформаційної частини, сайт передбачає особистий кабінет користувача, навчальний функціонал, адміністративну панель для керування контентом, а також можливість роботи із заявками на оплату курсів і вебінарів.

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність і термін окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

У цьому підрозділі розглянуто основні етапи технологічного процесу

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

створення вебсайту для психолога Ольги Демедчук. До складу робіт входять аналіз предметної області, формування вимог, проєктування структури сайту, розробка дизайну інтерфейсу, створення клієнтської та серверної частин, тестування, підготовка документації та розгортання програмного продукту.

Оскільки розроблений проєкт є вебсайтом з публічною частиною, особистим кабінетом користувача та адміністративною панеллю, трудомісткість робіт розподілено між кількома виконавцями: керівником проєкту, інженером першого рівня, інженером другого рівня та тестувальником. Для визначення загальної тривалості проведення робіт дані щодо витрат часу наведені в таблиці 4.1.

Таблиця 4.1 – Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Планування та аналіз	Кер. проєкту Рm	9
		Інженер (11)	9
2	Розробка технічного завдання	Кер. проєкту (Рm)	8
		Інженер (11)	5
3	Дизайн інтерфейсу	Інженер (11)	18
		Інженер (12)	26
4	Розробка функціоналу	Інженер (11)	50
5	Тестування та відладка	Тестувальник	11
6	Документування	Інженер (11)	4
7	Розгортання та підтримка	Інженер (12)	18
8	Управління проєктом	Кер. проєкту (Рm)	18
Разом			176

Сумарний час виконання операцій технологічного процесу становить 176 години.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У цьому підрозділі виконується розрахунок витрат, пов'язаних з оплатою праці виконавців, які беруть участь у створенні вебсайту для психолога Ольги Демедчук. До складу команди входять керівник проекту, інженер 1-го рівня, інженер 2-го рівня та тестувальник. Кожен виконавець залучається на певних етапах розробки відповідно до складності виконуваних робіт.

Розмір заробітної плати залежить від складності завдань, кваліфікації працівника, обсягу виконаних робіт і кількості фактично витрачених годин.

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн.}} = T_c * K_r \quad (4.1)$$

де: T_c – тарифна ставка, грн. (приймаємо для керівника проекту (Рм) – 600 грн./год, інженера (І2) – 420 грн./год.), інженера (І1) – 260 грн./год., тестувальник – 220 грн./год.; K_r – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

Керівника проекту (Рм) $Z_{\text{осн.}} = 35 * 600 = 21\,000$ грн.

Інженера (І2) $Z_{\text{осн.}} = 44 * 420 = 18\,480$ грн.

Інженера (І1) $Z_{\text{осн.}} = 86 * 260 = 22\,360$ грн.

Тестувальник $Z_{\text{осн.}} = 11 * 220 = 2\,420$ грн.

Сумарна основна заробітна плата становить

$$Z_{\text{осн.}} = 21\,000 + 18\,480 + 22\,360 + 2\,420 = 64\,260 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} * K_{\text{допл.}} \quad (4.2)$$

де: $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

Керівника проекту $Z_{\text{дод.2}} = 21\,000 * 0,1 = 2\,100$ грн.

Інженера (І2) $Z_{\text{дод.3}} = 18\,480 * 0,1 = 1\,848$ грн.

Інженера (І1) $Z_{\text{дод.4}} = 22\,360 * 0,1 = 2\,236$ грн.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

Тестувальник $Z_{\text{дод4}} = 2\,420 * 0,1 = 242$ грн.

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод}} = 2\,100 + 1\,848 + 2\,236 + 242 = 6\,426 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{\text{о.п.}}$) визначаються за формулою:

$$B_{\text{о.п.}} = Z_{\text{осн.}} + Z_{\text{дод.}} \quad (4.3)$$

$$B_{\text{о.п.}} = 64\,260 + 6\,426 = 70\,686 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$B_{\text{ЄСВ}} = B_{\text{о.п.}} * 0,22 \quad (4.4)$$

$$B_{\text{ЄСВ}} = 70\,686 * 0,22 = 15\,551 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п/ п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. 6 = 3+4+5
		Тарифна ставка, грн.	К-сть годин	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Кер. проекту (Pm)	600	35	21 000	2 100	5 082	28 182
2	Інженера (I2)	420	44	18 480	1 848	4 472	24 800
3	Інженера (I1)	260	86	22 360	2 236	5 411	30 007
4	Тестувальник	220	11	2 420	242	586	3 248
Разом				64 260	6 426	15 551	86 237

Отже, загальні витрати на оплату праці становлять 86 237 грн.

4.3 Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

обладнання визначаються за формулою:

$$Z_B = W * T * S \quad (4.5)$$

де: W – необхідна потужність, кВт; T – кількість годин роботи обладнання;
 S – вартість кіловат-години електроенергії (приймаємо 15,94 грн).

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні: комп'ютер – 0,82 кВт/год.

$$Z_{ек} = 0,82 * 176 * 15,94 = 2\,300 \text{ грн.}$$

Витрати на електроенергію становлять 2300 грн.

4.4 Розрахунок суми амортизаційних відрахувань вебсайту для психолога Ольги Демедчук

У процесі розробки програмного продукту використовується комп'ютерна техніка, яка поступово зношується. Для врахування цього зношування у собівартості робіт визначають амортизаційні відрахування.

Амортизація — це поступове перенесення вартості основних засобів на вартість виготовленого продукту. Для розробки вебсайту використовується один персональний комп'ютер вартістю 50 000 грн.

Комп'ютерна техніка належить до основних фондів, для яких мінімальний строк корисного використання становить 2 роки. Для визначення амортизаційних відрахувань використаємо формулу:

$$A = \frac{B_B * N_A}{100\%} * T, \quad (4.6)$$

де: A – амортизаційні відрахування за звітний період, грн.;

B_B – балансова вартість групи основних фондів на початок звітного періоду, грн.;

N_A – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ПК, вартістю 50000,00 грн., то сума амортизаційних відрахувань становитиме:

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

$$A = \frac{50\,000,00 * 0,04}{150} * 176 = 2\,347 \text{ грн.}$$

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з організацією процесу розробки, утриманням робочого місця, використанням офісного середовища, адмініструванням, зв'язком, доступом до мережі Інтернет, супровідними організаційними витратами та іншими витратами, які не входять безпосередньо до основної заробітної плати.

Для розробки вебсайту для психолога Ольги Демедчук накладні витрати доцільно прийняти на рівні 40 % від витрат на оплату праці без урахування ЄСВ, тобто від суми основної та додаткової заробітної плати.

$$H_B = V_{o.п.} * 0,2..0,6 \quad (4.7)$$

де: H_B – накладні витрати.

$$H_B = 70\,686 * 0,4 = 28\,274 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості вебсайту для психолога Ольги Демедчук

Для визначення повної собівартості розробки вебсайту необхідно об'єднати всі попередньо розраховані витрати: витрати на оплату праці з урахуванням ЄСВ, витрати на електроенергію, амортизаційні відрахування та накладні витрати.

Таблиця 4.3 – Кошторис витрат вебсайту для психолога Ольги Демедчук

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	86 237	72,4
2.	Витрати на електроенергію	2 300	1,9
3.	Амортизаційні відрахування	2 347	2,0
4.	Накладні витрати	28 274	23,7
5.	Собівартість	119 158	100

Собівартість (C_B) НДР розрахуємо за формулою:

$$C_B = V_{o.п} + V_{c.з} + Z_e + A + H_B \quad (4.8)$$

Отже, собівартість дорівнює $C_B = 86\,237 + 2\,300 + 2\,347 + 28\,274 = 119\,158$ грн.

4.7 Розрахунок ціни для вебсайту для психолога Ольги Демедчук

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка. Ціну робіт можна визначити за формулою:

$$Ц = C_B * (1 + P_{рен}) * (1 + ПДВ), \quad (4.9)$$

де: C_B – собівартість; $P_{рен}$ – рівень рентабельності; ПДВ – ставка податку на додану вартість.

$$Ц = 119\,158 \times (1 + 0,3) \times (1 + 0,2) = 185\,286 \text{ грн.}$$

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Плановий прибуток або грошовий потік визначається як різниця між ціною розробки та собівартістю:

$$\Gamma_{п} = Ц - C_B = 185\,286 - 119\,158 = 66\,128 \text{ грн.} \quad (4.10)$$

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$ЧТВ = -C_B + \sum_{i=1}^t \frac{\Gamma_{п}}{(1+i)^t}, \quad (4.11)$$

де: C_B – собівартість розробки; $\Gamma_{п}$ – грошовий потік за t – ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$ЧТВ = -119\,158 + \frac{66128}{(1 + 0,15)^1} + \frac{66128}{(1 + 0,15)^2} + \frac{66128}{(1 + 0,15)^3} = 31\,945 \text{ грн}$$

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

Якщо $ЧТВ \geq 0$, то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{OK} = T_{ПВ} + \frac{H_B}{Г_{ПР}} \quad (4.12)$$

де: $T_{ПВ}$ – період до повного відшкодування витрат, років; H_B – невідшкодовані витрати на початок року, грн.; $Г_{ПР}$ – грошовий потік на початок року, грн.

$$T_{OK} = 2 + \frac{11\,559}{66\,128} = 2,2 \text{ р.}$$

Основні техніко-економічні показники розробленого вебсайту наведено в таблиці 4.4.

Таблиця 4.4 – Техніко-економічні показники вебсайту для психолога Ольги Демедчук

№ п/п	Показник	Значення
1.	Собівартість, грн.	119 158
2.	Плановий прибуток або грошовий потік, грн.	66 128
3.	Ціна, грн.	185 286
4.	Чиста теперішня вартість, грн.	31 945
5.	Термін окупності, рік	2,2

Розраховані техніко-економічні показники підтверджують доцільність створення вебсайту для психолога Ольги Демедчук. Додатне значення чистої теперішньої вартості свідчить про економічний потенціал проекту, а термін окупності 2,2 року є прийнятним для веб-розробки такого типу.

Отже, проведені розрахунки підтверджують економічну доцільність розробки вебсайту для психолога Ольги Демедчук. Сайт забезпечує централізоване представлення послуг, курсів і вебінарів, спрощує взаємодію з користувачами та сприяє подальшому розвитку професійної діяльності психолога в Інтернеті.

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Інструкції з охорони праці, їх поділ

Охорона праці є важливою складовою організації безпечної трудової діяльності працівників на підприємстві. Вона охоплює систему правових, організаційних, технічних, санітарно-гігієнічних та профілактичних заходів, спрямованих на збереження життя, здоров'я та працездатності людини під час виконання професійних обов'язків. У сфері інформаційних технологій та веброзробки дотримання правил охорони праці також має велике значення, оскільки працівники тривалий час працюють за комп'ютерною технікою, електрообладнанням та іншими цифровими пристроями.

Основним документом, який регламентує безпечне виконання робіт, є інструкція з охорони праці. Вона містить перелік вимог, правил і рекомендацій щодо безпечної поведінки працівника під час виконання службових обов'язків, користування обладнанням та перебування на робочому місці. Інструкції створюються відповідно до чинного законодавства України та нормативних актів у сфері охорони праці.

Інструкції з охорони праці поділяються на кілька основних видів залежно від характеру робіт, посади працівника та умов виконання професійної діяльності. Найпоширенішими є:

Інструкції за професіями або посадами.

Такі інструкції розробляються окремо для кожної категорії працівників. Наприклад, для системного адміністратора, програміста, веброзробника або оператора комп'ютерного набору. У них визначаються вимоги безпеки перед початком роботи, під час роботи та після її завершення.

Інструкції за видами робіт.

Ці документи стосуються виконання певних робіт або використання конкретного обладнання. До них можуть належати інструкції під час роботи з електрообладнанням, офісною технікою, серверним обладнанням або мережевими

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

пристроями.

Тимчасові інструкції.

Такі інструкції вводяться на певний період часу у випадку впровадження нового обладнання, зміни технологічного процесу або проведення тимчасових робіт. Після завершення відповідних робіт вони можуть бути скасовані або оновлені.

Типові інструкції.

Типові інструкції створюються на державному або галузевому рівні та використовуються як основа для розробки локальних інструкцій на підприємстві. Вони враховують загальні вимоги безпеки для певного виду діяльності.

Локальні інструкції підприємства.

Такі інструкції розробляються безпосередньо організацією з урахуванням особливостей робочого середовища, технічного оснащення та специфіки діяльності працівників.

Кожна інструкція з охорони праці повинна містити чітку структуру. Зазвичай до неї входять загальні положення, вимоги безпеки перед початком роботи, під час роботи, після завершення роботи, а також порядок дій у разі виникнення аварійної ситуації. Працівник зобов'язаний ознайомитися з інструкцією під підпис перед початком виконання своїх обов'язків.

Особливе значення в ІТ-сфері має дотримання вимог ергономіки та організації робочого місця. Неправильне розташування монітора, недостатнє освітлення, тривала робота без перерв або порушення правил користування електроприладами можуть негативно впливати на стан здоров'я працівника. Саме тому інструкції повинні містити рекомендації щодо режиму праці та відпочинку, правильної постави, безпечного використання обладнання та підтримання належного мікроклімату в приміщенні.

Важливим елементом системи охорони праці є проведення інструктажів. На підприємствах проводяться вступний, первинний, повторний, позаплановий та цільовий інструктажі. Їх метою є ознайомлення працівників із правилами безпеки, попередження нещасних випадків та формування навичок правильної поведінки у

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підпис	Дата		

надзвичайних ситуаціях.

Таким чином, інструкції з охорони праці є невід'ємною частиною організації безпечної трудової діяльності. Вони забезпечують працівникам необхідний рівень обізнаності щодо правил безпеки, допомагають запобігати виробничим травмам та створюють безпечні умови праці під час виконання професійних обов'язків.

5.2 Електробезпека. Види ураження людини електричним струмом

Електробезпека є одним із найважливіших напрямів охорони праці, оскільки електричний струм становить серйозну небезпеку для життя та здоров'я людини. У сучасних умовах практично всі підприємства, офіси та робочі приміщення оснащені електрообладнанням, комп'ютерною технікою, мережевими пристроями та системами електроживлення. Саме тому працівники повинні дотримуватись правил безпечного користування електроприладами та знати основні заходи захисту від ураження струмом.

Електричний струм не має зовнішніх ознак, тому людина не може завчасно визначити його небезпеку без спеціальних приладів. Ураження виникає внаслідок проходження струму через організм людини при контакті з пошкодженим обладнанням, оголеними проводами або металевими частинами, що перебувають під напругою. Небезпека ураження залежить від сили струму, напруги, тривалості впливу, стану організму та умов навколишнього середовища.

Основними причинами ураження електричним струмом є:

- порушення правил експлуатації електрообладнання;
- використання несправних електроприладів;
- пошкодження ізоляції проводів;
- відсутність заземлення;
- недотримання правил техніки безпеки;
- робота з електрообладнанням в умовах підвищеної вологості.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

Ураження електричним струмом поділяються на кілька основних видів.

Термічне ураження.

Виникає внаслідок нагрівання тканин організму електричним струмом. Воно може супроводжуватись опіками шкіри, пошкодженням внутрішніх органів та перегріванням тканин. Особливо небезпечними є глибокі електричні опіки.

Біологічне ураження.

Електричний струм впливає на нервову систему та м'язи людини, викликаючи судоми, різкі скорочення м'язів, порушення роботи серця та органів дихання. У важких випадках можливе припинення серцебиття або дихання.

Електролітичне ураження.

Під дією струму змінюється хімічний склад крові та інших рідин організму. Це призводить до порушення нормального функціонування клітин і тканин.

Механічне ураження.

Внаслідок різкого скорочення м'язів або дії електричної дуги можуть виникати переломи, розриви тканин або травми від падіння.

Найнебезпечнішими наслідками ураження електричним струмом є електричний удар та електротравма. Електричний удар впливає на весь організм людини та може спричинити втрату свідомості, порушення серцевої діяльності або клінічну смерть. Електротравма зазвичай проявляється у вигляді місцевих ушкоджень — опіків, електричних знаків на шкірі чи пошкодження тканин.

Для запобігання нещасним випадкам необхідно дотримуватися правил електробезпеки:

- використовувати лише справне обладнання;
- не торкатися оголених проводів;
- не працювати з електроприладами вологими руками;
- регулярно перевіряти стан електропроводки;
- використовувати заземлення та автоматичні засоби захисту;
- вимикати обладнання після завершення роботи;
- не перевантажувати електромережу великою кількістю пристроїв.

					<i>2026.КВР.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

У разі ураження людини електричним струмом необхідно негайно вимкнути джерело живлення або відтягнути потерпілого від проводу за допомогою сухого предмета, який не проводить струм. Після цього слід оцінити стан постраждалого, перевірити наявність дихання та пульсу, викликати швидку медичну допомогу й за потреби розпочати серцево-легеневу реанімацію.

У приміщеннях, де використовується комп'ютерна техніка та серверне обладнання, важливо забезпечити належний технічний стан електромережі, справність систем охолодження та захист від короткого замикання. Працівники повинні проходити інструктажі з електробезпеки та знати порядок дій у разі аварійної ситуації.

Отже, електробезпека є важливою складовою охорони праці в сучасному цифровому середовищі. Дотримання правил безпечної роботи з електрообладнанням, проведення інструктажів та контроль технічного стану електромереж дозволяють значно знизити ризик виникнення нещасних випадків і забезпечити безпечні умови праці для працівників.

Особливу увагу слід приділяти профілактичним заходам. Перед початком роботи з комп'ютерною технікою необхідно переконатися у справності кабелів, розеток, подовжувачів і блоків живлення. Не допускається використання обладнання з пошкодженою ізоляцією, перегрітими елементами або нестабільним підключенням до електромережі. Також не рекомендується перевантажувати одну розетку великою кількістю пристроїв, оскільки це може призвести до перегрівання проводки та виникнення аварійної ситуації.

Для безпечної експлуатації електрообладнання важливо підтримувати порядок на робочому місці. Кабелі не повинні заважати пересуванню працівників або лежати в місцях, де їх можна випадково пошкодити. У разі виявлення запаху гару, іскріння, нестабільної роботи обладнання або перегрівання пристроїв необхідно негайно припинити роботу та повідомити відповідальну особу. Своєчасне виявлення таких ознак дає змогу запобігти пошкодженню техніки та знизити ризик ураження електричним струмом.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У межах кваліфікаційної роботи було розроблено вебсайт для психолога Ольги Демедчук для представлення професійної діяльності спеціаліста, інформування користувачів про послуги, курси, вебінари, відгуки та способи зв'язку. Сайт поєднує інформаційну частину, навчальний функціонал, особистий кабінет користувача та адміністративну панель.

Під час виконання роботи було проаналізовано предметну область, визначено основні вимоги до вебсайту та спроектовано його навігаційну структуру. До складу сайту увійшли головна сторінка, сторінки послуг, курсів, вебінарів, відгуків, контактів, авторизації, реєстрації та особистого кабінету. Така структура забезпечує зручний доступ до основної інформації та функцій системи.

Клієнтську частину реалізовано з використанням React, Vite, React Router DOM та Tailwind CSS. Компонентний підхід дав змогу створити єдиний стиль інтерфейсу, повторно використовувати елементи сторінок і забезпечити адаптивне відображення сайту на різних пристроях.

Серверну частину розроблено на основі Node.js та Express.js. Для зберігання даних використано MongoDB Atlas і Mongoose. База даних містить колекції для користувачів, послуг, курсів, модулів, уроків, вебінарів, замовлень, відгуків, фотографій та інших даних, необхідних для роботи сайту.

Також було реалізовано адміністративну панель, яка дозволяє керувати послугами, курсами, уроками, вебінарами, фотографіями, відгуками, користувачами та заявками на оплату. Це забезпечує підтримання сайту в актуальному стані без прямого редагування бази даних.

Після завершення розробки було проведено тестування основних функцій сайту. Перевірено роботу навігації, форм, авторизації, особистого кабінету, адміністративної панелі, API-запитів та адаптивної верстки. Результати тестування підтвердили, що розроблений вебсайт відповідає поставленому технічному завданню та може використовуватися для представлення діяльності психолога, взаємодії з користувачами та керування навчальним контентом.

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Марціяш Г.Я., Слободян Р.О. Методичні вказівки до виконання кваліфікаційної роботи для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки». Тернопіль : ВСП «ТФК ТНТУ ім. І. Пулюя», 2026, 48 с.
- 2) React Documentation. React. URL: <https://react.dev/> (дата звернення: 10.06.2026).
- 3) Vite Guide. Vite. URL: <https://vite.dev/guide/> (дата звернення: 10.06.2026).
- 4) React Router Documentation. React Router. URL: <https://reactrouter.com/> (дата звернення: 10.06.2026).
- 5) Node.js Documentation. Node.js. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 10.06.2026).
- 6) Express.js Documentation. Express. URL: <https://expressjs.com/> (дата звернення: 10.06.2026).
- 7) MongoDB Documentation. MongoDB. URL: <https://www.mongodb.com/docs/> (дата звернення: 10.06.2026).
- 8) Mongoose Documentation. Mongoose. URL: <https://mongoosejs.com/docs/> (дата звернення: 10.06.2026).
- 9) Cloudinary Documentation. Cloudinary. URL: <https://cloudinary.com/documentation> (дата звернення: 10.06.2026).
- 10) Vercel Documentation. Vercel. URL: <https://vercel.com/docs> (дата звернення: 10.06.2026).
- 11) Render Documentation. Render. URL: <https://render.com/docs> (дата звернення: 10.06.2026).
- 12) Postman API Platform. Postman. URL: <https://www.postman.com/> (дата звернення: 10.06.2026).
- 13) ChatGPT. OpenAI. URL: <https://chatgpt.com/> (дата звернення: 10.06.2026).

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

ДОДАТКИ

Додаток А. App.jsx

```
import { useEffect } from "react";
import { Navigate, Route, Routes, useLocation } from "react-router-dom";
import { ProtectedRoute } from "../components/auth/ProtectedRoute.jsx";
import { Layout } from "../components/Layout/Layout.jsx";
import { DashboardLayout } from "../components/dashboard/DashboardLayout.jsx";
import { LoginPage } from "../pages/auth/LoginPage.jsx";
import { RegisterPage } from "../pages/auth/RegisterPage.jsx";
import { ForgotPasswordPage } from "../pages/auth/ForgotPasswordPage.jsx";
import { ResetPasswordPage } from "../pages/auth/ResetPasswordPage.jsx";
import { BookingPage } from "../pages/booking/BookingPage.jsx";
import { ContactPage } from "../pages/ContactPage.jsx";
import { CoursesCatalogPage } from "../pages/courses/CoursesCatalogPage.jsx";
import { CourseDetailPage } from "../pages/courses/CourseDetailPage.jsx";
import { CourseLearnPage } from "../pages/courses/CourseLearnPage.jsx";
import { DashboardOverviewPage } from "../pages/dashboard/DashboardOverviewPage.jsx";
import { MyBookingsPage } from "../pages/dashboard/MyBookingsPage.jsx";
import { MyCoursesPage } from "../pages/dashboard/MyCoursesPage.jsx";
import { ProfilePage } from "../pages/dashboard/ProfilePage.jsx";
import { HomePage } from "../pages/HomePage.jsx";
import { NotFoundPage } from "../pages/NotFoundPage.jsx";
import { ServicesPage } from "../pages/ServicesPage.jsx";
import { TestimonialsPage } from "../pages/TestimonialsPage.jsx";
import { WebinarsPage } from "../pages/WebinarsPage.jsx";

function ScrollToTop() {
  const { pathname, hash } = useLocation();

  useEffect(() => {
    if (hash) {
```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

```

const id = hash.replace("#", "");
requestAnimationFrame(() => {
    document.getElementById(id)?.scrollIntoView({ behavior: "smooth",
block: "start" });
});
return;
}
window.scrollTo({ top: 0, behavior: "smooth" });
}, [pathname, hash]);

return null;
}

export default function App() {
return (
<Routes>
<Route element={<Layout />}>
<Route index element={<HomePage />} />
<Route path="services" element={<ServicesPage />} />
<Route path="courses" element={<CoursesCatalogPage />} />
<Route path="courses/:slug" element={<CourseDetailPage />} />
<Route path="courses/:slug/learn/*" element={<CourseLearnPage />}
/>

<Route path="webinars" element={<WebinarsPage />} />
<Route path="testimonials" element={<TestimonialsPage />} />
<Route path="contact" element={<ContactPage />} />
<Route path="booking" element={<BookingPage />} />

<Route path="login" element={<LoginPage />} />
<Route path="register" element={<RegisterPage />} />
<Route path="forgot-password" element={<ForgotPasswordPage />} />
<Route path="reset-password" element={<ResetPasswordPage />} />

<Route

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

```

    path="dashboard"
    element={
      <ProtectedRoute>
        <DashboardLayout />
      </ProtectedRoute>
    }
  >
  <Route index element={<DashboardOverviewPage />} />
  <Route path="courses" element={<MyCoursesPage />} />
  <Route path="bookings" element={<MyBookingsPage />} />
  <Route path="profile" element={<ProfilePage />} />
</Route>

  <Route path="404" element={<NotFoundPage />} />
  <Route path="*" element={<NotFoundPage />} />
</Route>
  <Route path="/home" element={<Navigate to="/" replace />} />
</Routes>
);
}

export { ScrollToTop }

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

Додаток Б. navigation.js

```
export const mainNav = [  
  { to: "/", label: "Головна", end: true },  
  { to: "/services", label: "Послуги" },  
  { to: "/courses", label: "Курси" },  
  { to: "/webinars", label: "Вебінари" },  
  { to: "/testimonials", label: "Відгуки" },  
  { to: "/contact", label: "Контакти" },  
];
```

```
export const footerNav = [  
  { to: "/services", label: "Послуги" },  
  { to: "/courses", label: "Курси" },  
  { to: "/webinars", label: "Вебінари" },  
  { to: "/testimonials", label: "Відгуки" },  
  { to: "/contact", label: "Контакти" },  
];
```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		76

Додаток В. HomePage.jsx

```
import { PageMeta } from "../components/seo/PageMeta.jsx";
import { pageMeta } from "../content/pageMeta.js";
import { AboutPreviewSection } from "../sections/AboutPreviewSection.jsx";
import { BenefitsSection } from "../sections/BenefitsSection.jsx";
import { CtaSection } from "../sections/CtaSection.jsx";
import      {      FeaturedCoursesSection      }      from
"../sections/FeaturedCoursesSection.jsx";
import { HeroSection } from "../sections/HeroSection.jsx";
import { HomeFaqSection } from "../sections/HomeFaqSection.jsx";
import      {      ServicesPreviewSection      }      from
"../sections/ServicesPreviewSection.jsx";
import      {      TestimonialsPreviewSection      }      from
"../sections/TestimonialsPreviewSection.jsx";

export function HomePage() {
  return (
    <>
      <PageMeta      title={pageMeta.home.title}
description={pageMeta.home.description} />
      <HeroSection />
      <AboutPreviewSection />
      <BenefitsSection />
      <ServicesPreviewSection limit={3} />
      <FeaturedCoursesSection limit={3} />
      <TestimonialsPreviewSection limit={8} />
      <HomeFaqSection />
      <CtaSection />
    </>
  );
}
```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
						77
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Г. ServicesPage.jsx

```
import { PageHero } from "../components/Layout/PageHero.jsx";
import { PageMeta } from "../components/seo/PageMeta.jsx";
import { pageMeta } from "../content/pageMeta.js";
import { CtaSection } from "../sections/CtaSection.jsx";
import { ServicesSection } from "../sections/ServicesSection.jsx";

export function ServicesPage() {
  return (
    <>
      <PageMeta title={pageMeta.services.title}
description={pageMeta.services.description} />
      <PageHero
        eyebrow="Послуги"
        title="Формати консультацій та терапії"
        subtitle="Індивідуальна робота, онлайн та офлайн – актуальний
перелік форматів для вашого запиту."
      />
      <ServicesSection showHeader={false} />
      <CtaSection className="border-t border-forest-500/20" />
    </>
  );
}
```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						78
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Д. BookingPage.jsx

```
import { useEffect, useMemo, useState } from "react";
import { useSearchParams } from "react-router-dom";
import toast from "react-hot-toast";
import { createBooking, fetchWebinars } from "../../api/bookingsApi.js";
import { ProtectedRoute } from "../../components/auth/ProtectedRoute.jsx";
import { PageHero } from "../../components/Layout/PageHero.jsx";
import { PageMeta } from "../../components/seo/PageMeta.jsx";
import { FormField, inputClassName } from
"../../components/ui/FormField.jsx";
import { Button } from "../../components/ui/Button.jsx";
import { LoadingState } from "../../components/ui/LoadingState.jsx";

function formatWhen(value) {
  if (!value) return "";
  return new Date(value).toLocaleString("uk-UA", {
    day: "2-digit",
    month: "long",
    year: "numeric",
    hour: "2-digit",
    minute: "2-digit",
  });
}

function isPaid(webinar) {
  return webinar?.isFree === false || Number(webinar?.price ?? 0) > 0;
}

function priceLabel(webinar) {
  if (!webinar || !isPaid(webinar)) return "Безкоштовно";
  return `${Number(webinar.price || 0).toLocaleString("uk-UA")} грн`;
}
```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

```

function WebinarRegistrationForm() {
  const [searchParams] = useSearchParams();
  const initialWebinarId = searchParams.get("webinar") || "";
  const [webinars, setWebinars] = useState([]);
  const [loading, setLoading] = useState(true);
  const [webinarId, setWebinarId] = useState(initialWebinarId);
  const [notes, setNotes] = useState("");
  const [submitting, setSubmitting] = useState(false);

  useEffect(() => {
    fetchWebinars()
      .then((data) => setWebinars(data.docs ?? []))
      .catch(() => toast.error("Не вдалося завантажити вебінари"))
      .finally(() => setLoading(false));
  }, []);

  const selected = useMemo(() => webinars.find((webinar) => webinar.id ===
webinarId), [webinars, webinarId]);

  async function handleSubmit(event) {
    event.preventDefault();
    if (!webinarId) {
      toast.error("Оберіть вебінар");
      return;
    }
    setSubmitting(true);
    try {
      const result = await createBooking({ webinarId, notes });
      if (result.paymentUrl) {
        toast.success("Заявку створено. Переходимо до оплати в Монобанк.",
{ duration: 3000 });
        window.location.href = result.paymentUrl;
        return;
      }
    }
  }
}

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		80

```

toast.success("Ви зареєстровані на безкоштовний вебінар.");
window.location.href = "/dashboard/bookings";
} catch (err) {
const body = err?.message || "";
const message =
  body.includes("409") || body.includes("already")
    ? "Ви вже записані на цей вебінар"
    : "Не вдалося створити запис";
toast.error(message);
setSubmitting(false);
}
}

return (
  <>
  <PageMeta
    title="Запис на вебінар"
    description="Оберіть вебінар, оплатіть участь і отримайте доступ
після підтвердження адміністратора."
  />
  <PageHero
    eyebrow="Вебінари"
    title="Запис на вебінар"
    subtitle="Для платних вебінарів після натискання кнопки відкриється
оплата Monobank. Посилання на Google Meet з'явиться після підтвердження
адміністратором."
  />
  <section className="py-12 md:py-16">
    <form onSubmit={handleSubmit} className="mx-auto max-w-xl space-y-5
brand-card p-6 md:p-8">
      {loading ? (
        <LoadingState label="Завантаження вебінарів..." />
      ) : webinars.length === 0 ? (
        <p className="text-sm text-gold-300/80">Наразі немає доступних

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

```

вебінарів для запису.</p>
    ) : (
      <>
        <FormField label="Вебінар" id="webinar">
          <select
            id="webinar"
            className={inputClassName}
            value={webinarId}
            onChange={(event) => setWebinarId(event.target.value)}
            required
          >
            <option value="">Оберіть вебінар</option>
            {webinars.map((webinar) => (
              <option key={webinar.id} value={webinar.id}>
                {webinar.title} - {formatWhen(webinar.date)} -
{priceLabel(webinar)}
              </option>
            ))}
          </select>
        </FormField>
        {selected && (
          <div className="rounded-lg border border-forest-500/30 bg-
forest-800/40 p-4 text-sm text-gold-300/90">
            <p className="font-semibold text-gold-
200">{formatWhen(selected.date)}</p>
            <p className="mt-1 text-gold-
400">{priceLabel(selected)}</p>
            {(selected.shortDescription || selected.description) && (
              <p className="mt-2">{selected.shortDescription ||
selected.description}</p>
            )}
            {selected.durationMinutes && (
              <p className="mt-2 text-gold-400/80">Тривалість:
{selected.durationMinutes} хв</p>

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82

```

        })
      </div>
    })
    <FormField label="Коментар" id="notes" hint="Необов'язково">
      <textarea
        id="notes"
        rows={3}
        className={inputClassName}
        value={notes}
        onChange={(event) => setNotes(event.target.value)}
      />
    </FormField>
    <Button type="submit" className="w-full justify-center"
disabled={submitting}>
      {submitting
        ? "Створення запису..."
        : isPaid(selected)
          ? "Записатися та перейти до оплати"
          : "Зареєструватися"}
    </Button>
  </>
  })
</form>
</section>
</>
);
}
export function BookingPage() {
  return (
    <ProtectedRoute>
      <WebinarRegistrationForm />
    </ProtectedRoute>
  );
}

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

Додаток Е. CourseDetailPage.jsx

```
import { useEffect, useState } from "react";
import { Link, useNavigate, useParams } from "react-router-dom";
import toast from "react-hot-toast";
import { fetchCourseBySlug, fetchCourseLearn } from
"../../api/coursesApi.js";
import { createCourseOrder } from "../../api/ordersApi.js";
import { CourseDescription } from
"../../components/courses/CourseDescription.jsx";
import { PageMeta } from "../../components/seo/PageMeta.jsx";
import { Button } from "../../components/ui/Button.jsx";
import { ErrorState, LoadingState } from
"../../components/ui/LoadingState.jsx";
import { useAuth } from "../../contexts/AuthContext.jsx";
import { courseDetailPath } from "../../lib/coursePaths.js";

function formatPrice(price) {
  if (!price) return "Безкоштовно";
  return `${price.toLocaleString("uk-UA")} грн`;
}

export function CourseDetailPage() {
  const { slug: slugParam } = useParams();
  const navigate = useNavigate();
  const { isAuthenticated } = useAuth();
  const [course, setCourse] = useState(null);
  const [enrolled, setEnrolled] = useState(false);
  const [orderStatus, setOrderStatus] = useState(null);
  const [loading, setLoading] = useState(true);
  const [purchasing, setPurchasing] = useState(false);

  const slug = slugParam?.trim() ?? "";
```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

```

useEffect(() => {
  let cancelled = false;
  async function load() {
    setLoading(true);
    try {
      const loadedCourse = await fetchCourseBySlug(slug);
      if (cancelled) return;
      setCourse(loadedCourse);
      if (loadedCourse && isAuthenticated) {
        try {
          const learnKey = loadedCourse.slug || loadedCourse.id;
          const learn = await fetchCourseLearn(learnKey);
          if (!cancelled) {
            setEnrolled(learn.enrolled);
            setOrderStatus(learn.order?.status ?? null);
          }
        } catch {
          /* Access check is optional on the public course page. */
        }
      }
    } catch {
      if (!cancelled) setCourse(null);
    } finally {
      if (!cancelled) setLoading(false);
    }
  }
  load();
  return () => {
    cancelled = true;
  };
}, [slug, isAuthenticated]);

async function handlePurchase() {
  if (!course) return;

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

```

if (!isAuthenticated) {
  navigate("/login", { state: { from: courseDetailPath(course) } });
  return;
}
setPurchasing(true);
try {
  const result = await createCourseOrder(course.id);
  if (result.alreadyOwned) {
    toast.success("У вас уже є доступ до цього курсу");
    navigate(`${courseDetailPath(course)}/learn`);
    return;
  }

  if (!result.paymentUrl) {
    throw new Error("payment_url_missing");
  }

  window.open(result.paymentUrl, "_blank", "noopener,noreferrer");
  toast.success("Заявку створено. Після оплати дочекайтеся підтвердження адміністратора.", {
    duration: 6000,
  });
  setOrderStatus("pending");
  navigate("/dashboard/courses");
} catch {
  toast.error("Не вдалося створити заявку на оплату");
} finally {
  setPurchasing(false);
}

if (loading) return <LoadingState label="Завантаження курсу..." />;
if (!course) return <ErrorState message="Курс не знайдено." />;

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		86

```

const learnPath = `${courseDetailPath(course)}/learn`;

return (
  <>
    <PageMeta title={` ${course.title} - курс` }
description={course.shortDescription || course.description?.slice(0, 160)}
/>
    <article className="mx-auto max-w-6xl px-4 py-12 sm:px-6 lg:px-8">
      <div className="grid gap-12 lg:grid-cols-2">
        <div className="overflow-hidden rounded-3xl border border-forest-
500/30 bg-forest-600/40">
          {course.thumbnailUrl ? (
            <img src={course.thumbnailUrl} alt="" className="aspect-video
w-full object-cover" />
          ) : (
            <div className="flex aspect-video items-center justify-center
text-sm font-semibold text-gold-400/45">
              Фото курсу
            </div>
          )}
        </div>
        <div>
          <p className="text-xs font-semibold uppercase tracking-widest
text-gold-400">Онлайн-курс</p>
          <h1 className="mt-2 font-display text-3xl text-gold-100 md:text-
4xl">{course.title}</h1>
          {course.shortDescription && (
            <p className="mt-4 text-lg leading-relaxed text-gold-
300/90">{course.shortDescription}</p>
          )}
          <p className="mt-4 text-2xl font-semibold text-gold-
400">{formatPrice(course.price)}</p>
          {course.durationHours && (
            <p className="mt-2 text-sm text-gold-400/70">Тривалість:

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		87

```

{course.durationHours} год</p>
    )}
    <div className="mt-8 flex flex-wrap gap-4">
      {enrolled ? (
        <Button to={learnPath}>Продовжити навчання</Button>
      ) : orderStatus === "pending" ? (
        <Button to="/dashboard/courses" variant="secondary">
          Очікує підтвердження оплати
        </Button>
      ) : (
        <Button type="button" onClick={handlePurchase}
disabled={purchasing}>
          {purchasing ? "Створення заявки..." : "Купити курс"}
        </Button>
      )}
      <Button to="/courses" variant="secondary">
        Усі курси
      </Button>
    </div>
    {orderStatus === "pending" && (
      <p className="mt-4 text-sm text-gold-400/80">
        Заявку надіслано. Після оплати через Монобанк дочекайтеся
        підтвердження адміністратора.
      </p>
    )}
    {!enrolled && !orderStatus && (
      <p className="mt-4 text-xs text-gold-400/60">
        Оплата відбувається через Монобанк. Після переказу
        адміністратор підтвердить доступ.
      </p>
    )}
  </div>
</div>

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		88

```

    {course.description && (
      <section className="mt-16 rounded-2xl border border-forest-500/30
bg-forest-600/40 p-6 md:p-10">
        <h2 className="font-display text-2xl text-gold-100">Про
курс</h2>
        <CourseDescription text={course.description} className="mt-6"
/>
      </section>
    )}

```

```

    {course.modulesOutline?.length > 0 && (
      <section className="mt-10 rounded-2xl border border-forest-500/30
bg-forest-600/40 p-6 md:p-10">
        <h2 className="font-display text-2xl text-gold-100">Структура
навчання</h2>
        <ol className="mt-6 space-y-6">
          {course.modulesOutline.map((mod) => (
            <li key={mod.id}>
              <h3 className="font-display text-lg text-gold-200">
                {mod.order !== null ? `${mod.order}. ` : ""}
                {mod.title}
              </h3>
              {mod.description && <p className="mt-2 text-sm leading-
relaxed text-gold-300/85">{mod.description}</p>}
              {mod.lessons?.length > 0 && (
                <ul className="mt-3 space-y-1.5 border-1 border-gold-
400/25 pl-4 text-sm text-gold-300/80">
                  {mod.lessons.map((lesson) => (
                    <li key={lesson.id}>
                      {lesson.order !== null ? `${lesson.order}. ` : ""}
                      {lesson.title}
                    </li>
                  ))}
                </ul>
              )}
            </li>
          )}
        </ol>

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		89

```

        })
    </li>
    )})
</ol>
</section>
})

{course.program && (
    <section className="mt-10 rounded-2xl border border-forest-500/30
bg-forest-600/40 p-6 md:p-8">
        <h2 className="font-display text-2xl text-gold-100">Програма
курс</h2>
        <pre className="mt-4 whitespace-pre-wrap font-sans text-sm
leading-relaxed text-gold-300/90">
            {course.program}
        </pre>
    </section>
})

<p className="mt-8">
    <Link to="/courses" className="text-sm text-gold-400 hover:text-
gold-200">
        Назад до каталогу
    </Link>
</p>
</article>
</>
);
}

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		90

Додаток Ж. payloadClient.js

```
export function apiBaseUrl() {
  const raw = import.meta.env.VITE_API_URL;
  if (raw == null || String(raw).trim() === "") return "";
  return String(raw).replace(/\/$/, "");
}

const AUTH_TOKEN_STORAGE_KEY = "psytho_auth_token";

export function getAuthToken() {
  return localStorage.getItem(AUTH_TOKEN_STORAGE_KEY) || "";
}

export function setAuthToken(token) {
  if (token) {
    localStorage.setItem(AUTH_TOKEN_STORAGE_KEY, token);
  } else {
    localStorage.removeItem(AUTH_TOKEN_STORAGE_KEY);
  }
}

export async function fetchJSON(path, options = {}) {
  const { headers: optionHeaders, ...rest } = options;
  const base = apiBaseUrl();
  const url = path.startsWith("http") ? path :
  `${base}${path.startsWith("/") ? path : `/${path}`}`;
  const token = getAuthToken();
  const headers = { accept: "application/json", ...optionHeaders };
  if (token && !headers.Authorization && !headers.authorization) {
    headers.Authorization = `Bearer ${token}`;
  }
  const res = await fetch(url, {
    credentials: "include",
    ...rest,
  });
}
```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		91

```
headers,  
});  
if (!res.ok) {  
  const text = await res.text().catch(() => "");  
  throw new Error(`Request failed (${res.status}): ${text ||  
res.statusText}`);  
}  
return res.json();  
}
```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		92

Додаток II. server.js

```
import express from "express";
import cors from "cors";
import helmet from "helmet";
import cookieParser from "cookie-parser";
import pinoHttp from "pino-http";
import fs from "fs";
import path from "path";
import { fileURLToPath } from "url";
import rateLimit from "express-rate-limit";

import { env } from "./config/env.js";
import { connectDatabase } from "./config/db.js";
import { corsOriginCheck } from "./config/cors.js";
import { logger } from "./logger.js";
import apiRoutes from "./routes/index.js";
import { errorHandler, notFoundHandler } from
"./middleware/errorHandler.js";

const __dirname = path.dirname(fileURLToPath(import.meta.url));

const app = express();

if (env.trustProxy) {
  app.set("trust proxy", 1);
}

app.disable("x-powered-by");
app.use(pinoHttp({ logger }));
app.use(
  cors({
    origin: corsOriginCheck,
    credentials: true,
```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		93

```

    }),
  );
  app.use(helmet({ crossOriginResourcePolicy: { policy: "cross-origin" } }));
  app.use(cookieParser());
  app.use(express.json({ limit: "15mb" }));
  app.use(express.urlencoded({ extended: true }));

  app.use(
    rateLimit({
      windowMs: 60 * 1000,
      max: Number(process.env.API_RATE_LIMIT_MAX) || 120,
      standardHeaders: true,
      legacyHeaders: false,
    }),
  );

  const mediaLegacyPath = path.resolve(__dirname, "../media");

  app.use("/media", express.static(mediaLegacyPath));

  app.use("/api", apiRoutes);

  const adminPath = path.resolve(__dirname, "../admin/public");
  app.use("/admin", express.static(adminPath));
  app.get("/admin", (_req, res) => {
    res.sendFile(path.join(adminPath, "index.html"));
  });
  app.get("/admin/*", (_req, res) => {
    res.sendFile(path.join(adminPath, "index.html"));
  });

  const frontendPath = path.resolve(__dirname, "../..../frontend/dist");
  const frontendIndexPath = path.join(frontendPath, "index.html");
  const hasFrontendBuild = fs.existsSync(frontendIndexPath);

```

					<i>2026.KBP.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		94

```

if (hasFrontendBuild) {
  app.use(express.static(frontendPath));
  app.get("*", (req, res, next) => {
    if (req.path.startsWith("/api") || req.path.startsWith("/admin") ||
req.path.startsWith("/media")) {
      return next();
    }
    return res.sendFile(frontendIndexPath, (err) => {
      if (err) next(err);
    });
  });
} else {
  app.get("/", (_req, res) => {
    res.json({
      ok: true,
      service: "psytho-api",
      message: "Backend is running. Frontend build was not found in this
Render service.",
    });
  });
}

app.use(notFoundHandler);
app.use(errorHandler);
const VITE_DEV_PROXY_PORT = 3002;
async function start() {
  await connectDatabase();
  const server = app.listen(env.port, () => {
    logger.info(`API: http://localhost:${env.port}`);
    logger.info(`Admin: http://localhost:${env.port}/admin`);
    if (env.nodeEnv !== "production" && env.port !== VITE_DEV_PROXY_PORT) {
      logger.warn(
        { port: env.port, viteProxyPort: VITE_DEV_PROXY_PORT },

```

					<i>2026.KBP.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		95

```

        `PORT in .env (${env.port}) does not match Vite proxy
(:${VITE_DEV_PROXY_PORT}); frontend may receive 502. Set
PORT=${VITE_DEV_PROXY_PORT} in backend/.env`,
    );
    console.warn(
        `[psytho] Warning: backend runs on :${env.port}, while Vite proxy
targets :${VITE_DEV_PROXY_PORT}; browser may receive 502 Bad Gateway`,
    );
}
});
server.on("error", (err) => {
    if (err.code === "EADDRINUSE") {
        logger.fatal(
            { port: env.port, code: err.code },
            `Port ${env.port} is already in use. Stop another process: netstat
-ano | findstr :${env.port}`,
        );
        console.error(`\nEADDRINUSE - port ${env.port} is already in use.\n`);
        process.exit(1);
    }
    logger.fatal({ err }, "HTTP server error");
    process.exit(1);
});
}
process.on("unhandledRejection", (reason) => {
    logger.error({ err: reason }, "Unhandled promise rejection");
    if (reason instanceof Error && reason.stack) console.error(reason.stack);
});

start().catch((err) => {
    logger.fatal({ err, stack: err.stack }, "Start failed");
    if (err?.stack) console.error(err.stack);
    process.exit(1);
});

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		96

Додаток К. courseRoutes.js

```
import { Router } from "express";
import { asyncHandler } from "../utils/asyncHandler.js";
import * as course from "../controllers/courseController.js";
import { optionalAuth } from "../middleware/auth.js";

const router = Router();

router.get("/", asyncHandler(course.listCoursesHandler));
router.get("/:slug/learn", optionalAuth, asyncHandler(course.learnCourse));
router.get("/:slug", asyncHandler(course.getCourseHandler));

export default router;
```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
						97
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Л. admin.js

```
import {
  PAYMENT_REQUESTS_KEY,
  REVIEWS_MODERATION_KEY,
  RESOURCE_SCHEMAS,
  RESOURCE_KEYS,
} from "./admin-fields.js";
import {
  escapeHtml,
  formatTableCell,
  parseFormData,
  renderFormFields,
} from "./admin-form.js";
import { clearLookupCache, loadLookupsForSchema } from "./admin-lookups.js";

const RESOURCES = RESOURCE_KEYS.map((key) => ({
  key,
  label: RESOURCE_SCHEMAS[key].label,
  ...RESOURCE_SCHEMAS[key],
})));

let currentResource = PAYMENT_REQUESTS_KEY;
let currentRows = [];
let editingId = null;
let currentSearch = "";
const MAX_PHOTO_BYTES = 8 * 1024 * 1024;
const ALLOWED_PHOTO_TYPES = new Set(["image/jpeg", "image/png",
"image/webp", "image/gif"]);

const $ = (sel) => document.querySelector(sel);

async function api(path, options = {}) {
  const res = await fetch(`/api/admin${path}`, {
```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
						98
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    credentials: "include",
    headers: { "Content-Type": "application/json", Accept:
"application/json" },
    ...options,
  });
  const data = await res.json().catch(() => ({}));
  if (!res.ok) throw new Error(data.message || data.error || res.statusText);
  return data;
}

function renderImagePreview(root, url, emptyText = "Фото не вибрано") {
  const preview = root.querySelector(".image-uploader-preview");
  if (!preview) return;
  if (!url) {
    preview.className = "image-uploader-preview image-uploader-preview-
empty";
    preview.textContent = emptyText;
    return;
  }
  preview.className = "image-uploader-preview";
  preview.innerHTML = `})

```

					<i>2026.KBP.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		99

```

const dataUrl = await readFileAsDataURL(file);
const res = await fetch("/api/admin/photos/upload", {
  method: "POST",
  credentials: "include",
  headers: { "Content-Type": "application/json", Accept:
"application/json" },
  body: JSON.stringify({ filename: file.name, dataUrl }),
});
const data = await res.json().catch(() => ({}));
if (!res.ok) throw new Error(data.message || data.error || res.statusText);
if (!data.imageUrl) throw new Error("Cloudinary не повернув URL");
return data.imageUrl;
}

```

```

function initPhotoControls(scope) {
  scope.querySelectorAll("[data-photo-upload]").forEach((root) => {
    const fileInput = root.querySelector(".photo-file-input");
    const urlInput = root.querySelector(".photo-upload-url");
    const status = root.querySelector(".photo-upload-status");

    fileInput?.addEventListener("change", async () => {
      const file = fileInput.files?.[0];
      fileInput.value = "";
      if (!file) return;
      if (!file.type.startsWith("image/")) {
        alert("Оберіть файл зображення.");
        return;
      }
      if (!ALLOWED_PHOTO_TYPES.has(file.type)) {
        alert("Підтримуються тільки JPG, PNG, WebP або GIF.");
        return;
      }
      if (file.size > MAX_PHOTO_BYTES) {
        alert("Фото має бути не більше 8 МБ.");
      }
    });
  });
}

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		100

```

    return;
}

if (status) status.textContent = "Завантажуємо фото у Cloudinary...";
try {
    const imageUrl = await uploadPhotoFile(file);
    urlInput.value = imageUrl;
    renderImagePreview(root, imageUrl, "Фото ще не завантажено");
    if (status) status.textContent = "Фото завантажено. У базі буде збережено тільки URL.";
} catch (err) {
    if (status) status.textContent = "Не вдалося завантажити фото.";
    alert(err.message || "Помилка завантаження фото");
}
});
});

scope.querySelectorAll("[data-photo-url]").forEach((root) => {
    const input = root.querySelector(".photo-url-input");
    input?.addEventListener("input", () => renderImagePreview(root, input.value.trim(), "URL ще не вказано"));
});

scope.querySelectorAll("[data-photo-select]").forEach((root) => {
    const select = root.querySelector(".photo-select-input");
    select?.addEventListener("change", () => {
        const option = select.options[select.selectedIndex];
        renderImagePreview(root, option?.dataset?.imageUrl || "", "Фото не вибрано");
    });
});
}

async function checkSession() {

```

					<i>2026.KBP.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		101

```

try {
    const me = await fetch("/api/users/me", { credentials: "include"
}).then((r) => r.json());
    if (me?.user?.role === "admin") {
        showDashboard(me.user);
        return true;
    }
} catch {
    /* ignore */
}
return false;
}

function showLogin() {
    $("#login-view").hidden = false;
    $("#dashboard-view").hidden = true;
}

function showDashboard(user) {
    $("#login-view").hidden = true;
    $("#dashboard-view").hidden = false;
    $("#admin-email").textContent = user.email;
    renderTabs();
    loadStats();
    loadTable();
}

function getSchema() {
    return RESOURCE_SCHEMAS[currentResource];
}

function renderTabs() {
    const nav = $("#resource-tabs");
    nav.innerHTML = "";

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		102

```

RESOURCES.forEach((r) => {
  const btn = document.createElement("button");
  btn.type = "button";
  btn.textContent = r.label;
  btn.className = r.key === currentResource ? "active" : "";
  btn.onclick = () => {
    currentResource = r.key;
    currentSearch = "";
    $("#search-input").value = "";
    clearLookupCache();
    renderTabs();
    loadTable();
  };
  nav.appendChild(btn);
});
}

async function loadStats() {
  try {
    const s = await api("/stats");
    $("#stats").innerHTML = `
      <div class="stat-card"><strong>${s.users} ??
0}</strong>Користувачі</div>
      <div class="stat-card"><strong>${s.courses} ?? 0}</strong>Курси</div>
      <div class="stat-card"><strong>${s.photos} ?? 0}</strong>Фото</div>
      <div class="stat-card"><strong>${s.paidOrders} ??
0}</strong>Оплачені</div>
      <div class="stat-card"><strong>${s.pendingOrders} ?? 0}</strong>Оплата
очікує</div>
    `;
  } catch {
    $("#stats").innerHTML = "";
  }
}

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		103

```

function refLabel(v) {
  if (!v || typeof v !== "object") return null;
  if (v.title) return v.title;
  if (v.email) {
    const name = [v.firstName, v.lastName].filter(Boolean).join(" ").trim();
    return name ? `${name} (${v.email})` : v.email;
  }
  if (v.course?.title) return v.course.title;
  if (v.webinar?.title) return v.webinar.title;
  if (v.slug) return v.slug;
  return null;
}

```

```

function formatOrderDate(iso) {
  if (!iso) return "-";
  return new Date(iso).toLocaleString("uk-UA", {
    day: "2-digit",
    month: "2-digit",
    year: "numeric",
    hour: "2-digit",
    minute: "2-digit",
  });
}

```

```

function userDisplayName(user) {
  if (!user || typeof user !== "object") return "-";
  const name = [user.firstName, user.lastName].filter(Boolean).join(" ").trim();
  return name || user.email || "-";
}

```

```

function statusBadge(status) {
  const labels = { pending: "Очікує", paid: "Оплачено", rejected: "Відхилено"

```

					<i>2026.KBP.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		104

```

};
return ``;
}

function orderItemTitle(row) {
  if (row.course?.title) return row.course.title;
  if (row.webinar?.title) return row.webinar.title;
  return "-";
}

async function loadPaymentRequests() {
  const data = await api("/payment-requests");
  const rows = data.docs || [];
  $("#payments-body").innerHTML = rows
    .map((row) => {
      const actions =
        row.status === "pending"
          ? `

```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		105

```

    })
    .join("");

$("#payments-body").querySelectorAll("[data-approve]").forEach((btn) => {
    btn.onclick = () => approvePayment(btn.dataset.approve);
});
$("#payments-body").querySelectorAll("[data-reject]").forEach((btn) => {
    btn.onclick = () => rejectPayment(btn.dataset.reject);
});
}

async function approvePayment(id) {
    if (!confirm("Підтвердити оплату?")) return;
    await api(`/payment-requests/${id}/approve`, { method: "PATCH", body:
JSON.stringify({}) });
    window.location.reload();
}

async function rejectPayment(id) {
    const adminComment = prompt("Причина відхилення:");
    if (!adminComment?.trim()) return;
    await api(`/payment-requests/${id}/reject`, {
        method: "PATCH",
        body: JSON.stringify({ adminComment: adminComment.trim() }),
    });
    window.location.reload();
}

async function loadReviewsModeration() {
    const data = await api("/reviews");
    const rows = data.docs || [];
    $("#reviews-body").innerHTML = rows
        .map((row) => {
            const text = escapeHtml((row.text || "").slice(0, 120) + ((row.text ||

```

					<i>2026.КВР.122.421.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		106

```

"".length > 120 ? "... " : ""));
    const status = row.isApproved
      ? '<span class="status-badge status-paid">Опубліковано</span>'
      : '<span class="status-badge status-pending">На модерації</span>';
    const actions = row.isApproved
      ? `<button type="button" class="btn-reject" data-delete-
review="${row.id}">Видалити</button>`
      : `<button type="button" class="btn-approve" data-approve-
review="${row.id}">Схвалити</button>
      <button type="button" class="btn-reject" data-delete-
review="${row.id}">Видалити</button>`;
    return `<tr>
      <td>${escapeHtml(row.name ?? "-")}</td>
      <td>${text}</td>
      <td>${row.rating ? "★".repeat(row.rating) : "-"}</td>
      <td>${formatOrderDate(row.createdAt)}</td>
      <td>${status}</td>
      <td class="row-actions">${actions}</td>
    </tr>`;
  })
  .join("");

  $("#reviews-body").querySelectorAll("[data-approve-
review]").forEach((btn) => {
    btn.onclick = () => approveReviewAdmin(btn.dataset.approveReview);
  });
  $("#reviews-body").querySelectorAll("[data-delete-review]").forEach((btn)
=> {
    btn.onclick = () => deleteReviewAdmin(btn.dataset.deleteReview);
  });
}

async function approveReviewAdmin(id) {
  await fetch(`/api/reviews/${id}`, {

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		107

```

method: "PATCH",
credentials: "include",
headers: { "Content-Type": "application/json", Accept:
"application/json" },
});
await loadReviewsModeration();
}

```

```

async function deleteReviewAdmin(id) {
if (!confirm("Видалити відгук?")) return;
await fetch(`/api/reviews/${id}`, {
method: "DELETE",
credentials: "include",
headers: { Accept: "application/json" },
});
await loadReviewsModeration();
}

```

```

function setViewMode({ payments = false, reviews = false } = {}) {
const special = payments || reviews;
$("#payments-panel").hidden = !payments;
$("#reviews-panel").hidden = !reviews;
$("#crud-table").hidden = special;
$("#crud-toolbar").hidden = special;
}

```

```

async function loadTable() {
const schema = getSchema();
if (schema?.special === "payments") {
setViewMode({ payments: true });
await loadPaymentRequests();
return;
}
if (schema?.special === "reviews") {

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		108

```

    setViewMode({ reviews: true });
    await loadReviewsModeration();
    return;
}

setViewMode({});
const query = currentSearch ?
`?limit=100&q=${encodeURIComponent(currentSearch)}` : "?limit=100";
const data = await api(`/${currentResource}${query}`);
currentRows = data.docs || [];

const cols = [...(schema.tableColumns || []), "actions"];
const colLabels = {
    imagePreview: "Фото",
    imageUrl: "URL",
    slug: "Slug",
    accessType: "Тип",
    isPublished: "Статус",
    isActive: "Активний",
    isFree: "Тип",
    paymentStatus: "Статус оплати",
    entityType: "Тип",
    contentType: "Тип",
    progressPercent: "Прогрес",
    course: "Курс",
    module: "Модуль",
    user: "Користувач",
    service: "Послуга",
    webinar: "Вебінар",
    createdAt: "Створено",
    updatedAt: "Оновлено",
    date: "Дата",
    timelineStatus: "Стан",
    audience: "Для кого",

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		109

```

    role: "Роль",
    firstName: "Ім'я",
    lastName: "Прізвище",
    email: "Email",
    title: "Назва",
    amount: "Сума",
    price: "Ціна",
    status: "Статус",
    actions: "Дії",
  };
  const fieldLabels = Object.fromEntries((schema.fields || []).map((field)
=> [field.name, field.label]));

  $("#table-head").innerHTML = `|  |  |
| --- | --- |
|${cols.map((c) => ` ${colLabels[c] || fieldLabels[c] || c}</th>`).join("")}</tr>`;   $("#table-body").innerHTML = currentRows     .map((row) => {       const cells = cols         .filter((c) => c !== "actions")         .map((c) => ` ${formatTableCell(row, c, refLabel)}</td>`)         .join("");       return ` | |

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		110

```

async function openEdit(id) {
  editingId = id || null;
  const schema = getSchema();
  let row = {};

  if (id) {
    try {
      const { doc } = await api(`/${currentResource}/${id}`);
      row = doc || {};
    } catch {
      row = currentRows.find((r) => r.id === id) || {};
    }
  }

  $("#dialog-title").textContent = id ? "Редагування" : "Новий запис";
  $("#delete-btn").hidden = !id;
  $("#edit-fields").innerHTML = '<p class="muted">Завантаження
форми...</p>';
  $("#edit-dialog").showModal();

  let lookups = {};
  try {
    lookups = await loadLookupsForSchema(schema, api);
  } catch (err) {
    $("#edit-fields").innerHTML = '<p
class="error">${escapeHtml(err.message)}</p>`;
    return;
  }

  $("#edit-fields").innerHTML = renderFormFields(schema, row, lookups);
  $("#edit-fields").querySelectorAll(".toggle-input").forEach((input) => {
    input.addEventListener("change", () => {
      const label = input.closest(".field-toggle")?.querySelector(".toggle-
label");

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		111

```

        if (label) label.textContent = input.checked ? "Так" : "Hi";
    });
});
initPhotoControls($("#edit-fields"));
}

async function saveEdit(e) {
    e.preventDefault();
    const schema = getSchema();
    const fd = new FormData($("#edit-form"));
    const body = parseFormData(schema, fd);

    try {
        if (editingId) {
            await api(`/${currentResource}/${editingId}`, { method: "PATCH", body:
JSON.stringify(body) });
        } else {
            await api(`/${currentResource}`, { method: "POST", body:
JSON.stringify(body) });
        }
        $("#edit-dialog").close();
        clearLookupCache();
        await loadTable();
        await loadStats();
    } catch (err) {
        alert(err.message || "Помилка збереження");
    }
}

async function deleteRow() {
    if (!editingId || !confirm("Видалити запис?")) return;
    await api(`/${currentResource}/${editingId}`, { method: "DELETE" });
    $("#edit-dialog").close();
    clearLookupCache();
}

```

					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		112

```

    await loadTable();
  }

  let searchTimer = null;
  $("#search-input").addEventListener("input", (event) => {
    currentSearch = event.target.value.trim();
    clearTimeout(searchTimer);
    searchTimer = setTimeout(loadTable, 250);
  });

  $("#login-form").onsubmit = async (e) => {
    e.preventDefault();
    const fd = new FormData(e.target);
    $("#login-error").hidden = true;
    try {
      await api("/login", {
        method: "POST",
        body: JSON.stringify({ email: fd.get("email"), password:
fd.get("password") }),
      });
      const me = await fetch("/api/users/me", { credentials: "include"
}).then((r) => r.json());
      showDashboard(me.user);
    } catch {
      $("#login-error").textContent = "Невірний email або пароль";
      $("#login-error").hidden = false;
    }
  };

  $("#logout-btn").onclick = async () => {
    await fetch("/api/users/logout", { method: "POST", credentials: "include"
});
    showLogin();
  };

```

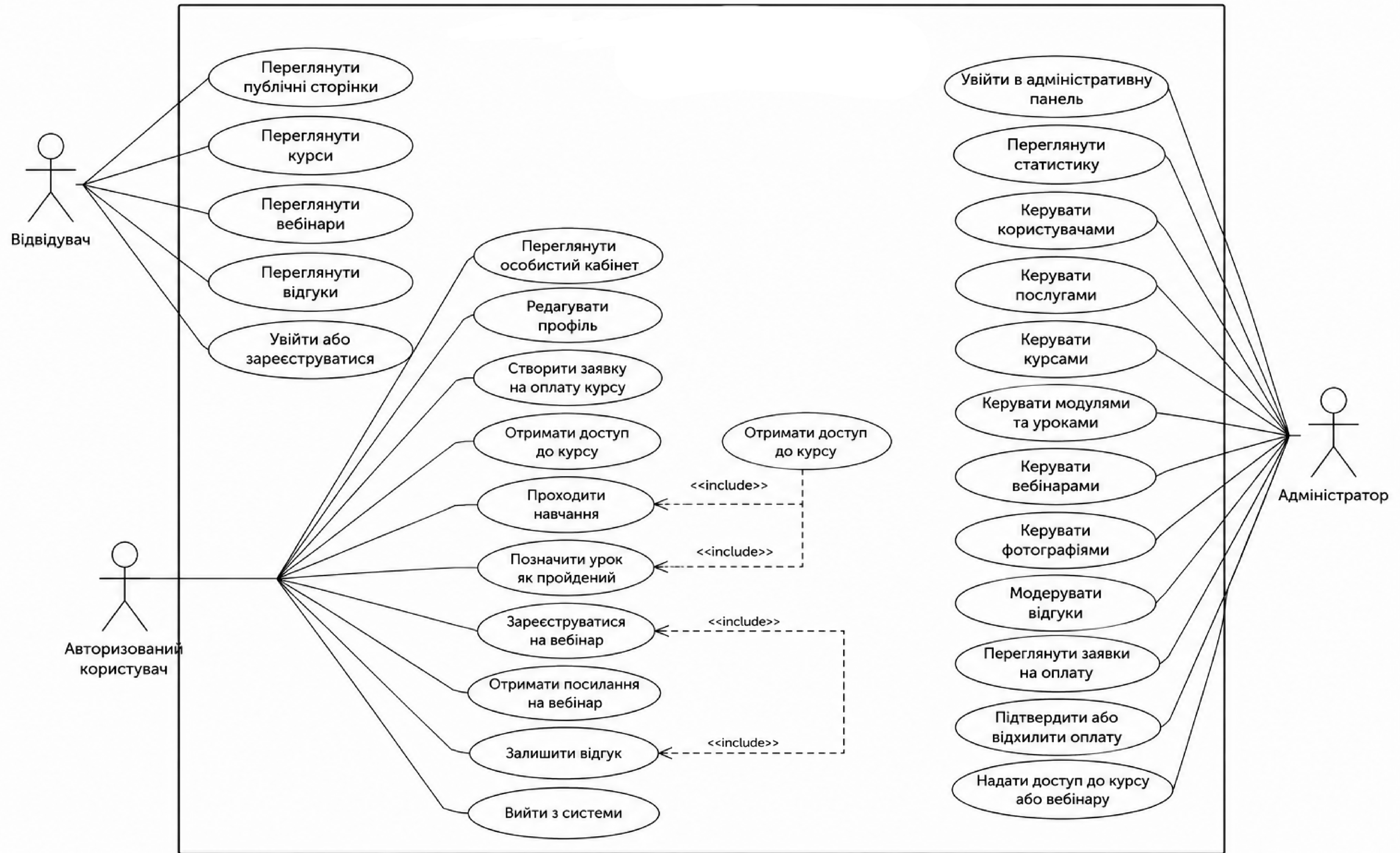
					2026.КВР.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		113

```
$("#refresh-btn").onclick = () => {
    loadTable();
    loadStats();
};
$("#create-btn").onclick = () => openEdit(null);
$("#edit-form").onsubmit = saveEdit;
$("#cancel-btn").onclick = () => $("#edit-dialog").close();
$("#delete-btn").onclick = deleteRow;

checkSession().then((ok) => {
    if (!ok) showLogin();
});
```

					2026.KBP.122.421.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		114

UML-діаграма варіантів використання програми

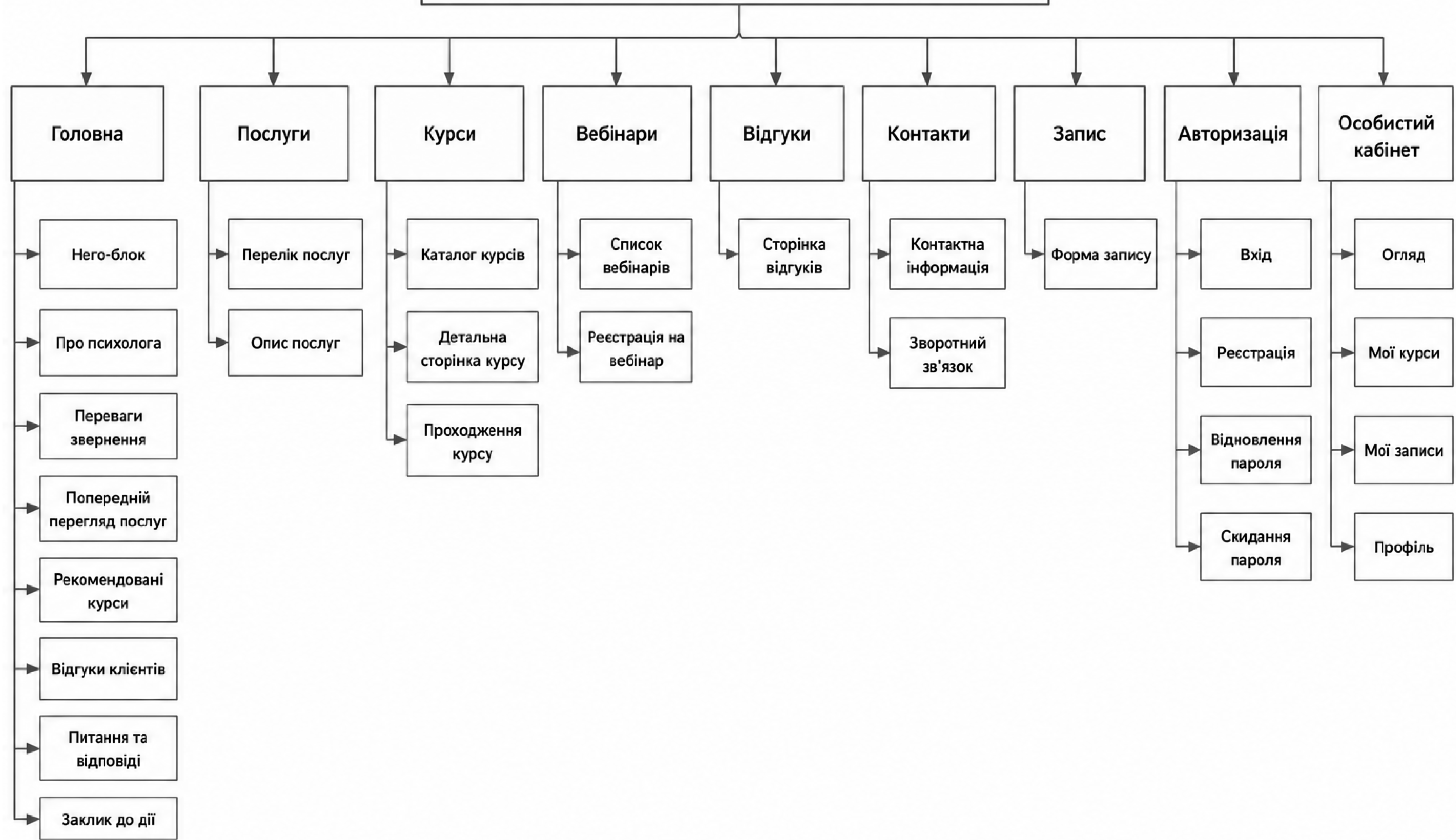


Перш. застос.
Стор. №
Підп. і дата
Зам. №, №
№, №, №

				2026.KBP.122.421.15.00.00 ДВ		
Зм.	Арх.	№ док.	Підп.	Дата	Розробка дизайну та навігаційної структури вебсайту для психолога Ольги Демедчук	
Розроб.	Мацюк Д. В.				Лист	Маса
Перев.	Марціняк Г. Я.				Архив	Архив
Г. контр.					Архив	1
Рецензент					ВП ТФК ТНТУ КН-421	
Нхонтр.	Приймак В.А.				м. Тернопіль	
Затв.					Формат А1	

Структурна схема головної сторінки сайту

Розробка дизайну та навігаційної структури вебсайту для психолога Ольги Демедчук



Перш. запис. Справ. № Підп. і дата. Підп. і дата. Підп. і дата. Підп. і дата. Підп. і дата. Підп. і дата.

2026.KBP.122.421.15.00.00 CC					Лист	Маса	Масштаб
Зм.	Арх.	№ док.	Підп.	Дата	Розробка дизайну та навігаційної структури вебсайту для психолога Ольги Демедчук Схема структурна		
Розроб.	Можук Д.В.						
Перев.	Марціняк Г.Я.				Архив	Архив	1
І.контр.							
Рецензент							
І.контр.	Приймак В.А.				ВСП ТФК ТНТУ КН-421		
Затв.					м. Тернопіль		
					Формат А1		

Таблиця техніко-економічних показників

№	Показник	Одиниця вимірюван	Значення
1	Платформа	-	Visual Studio Code
2	Інтерфейс	-	ІнтуїТИВНИЙ
3	Frontend	-	React
4	Backend	-	Node.js / Express.js
5	База даних	-	MongoDB Atlas
6	Загальний розмір програми	МБ	61,4
7	Собівартість	грн	119158
8	Плановий річний грошовий потік	грн	80000
9	Ціна	грн	185286
10	Чиста теперішня вартість	грн	63498
11	Термін окупності	рік	1,8

				2026.KBP.122.421.15.00.00 ТБ			
Зм.	Арк.	№ док.	Підп.	Дата	Лист	Маса	Масштаб
Розроб.	Мацюк Д. В.						
Перев.	Марціняк Г. Я.						
Г. контр.					Аркциш	Аркциш	1
Рецензент					ВПТ ТФК ТНТУ КН-4.21		
Нхонтр.	Приймак В.А.				м. Тернопіль		
Затв.					Формат А1		