

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 вебзастосунку для створення та реалізації
індивідуального інтер'єру та екстер'єру «Interiorix»**

Виконав: студент IV курсу, групи КН-421
спеціальності: 122 Комп'ютерні науки

Іван БІДЛОВСЬКИЙ

Керівник Галина МАРЦІЯШ

Рецензент Богдана МЛИНКО
(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 МАРЦЯШ

« 02 » березня 2026 року

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

Бідловський Іван Юрійович

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебзастосунку для створення та реалізації індивідуального інтер'єру та екстер'єру «Interiorix»

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: TypeScript; фреймворки: NestJS, Node.js; бібліотека React, Tailwind CSS; база даних MongoDB, 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 проєкту

2.1 Розробка структури вебзастосунку

2.2 Створення та верстка сторінок вебзастосунку

2.3 Розробка структури бази даних

2.4 Програмування вебзастосунку

2.4.1 Написання клієнтської частини

2.4.2 Написання серверної частини

2.5 Тестування вебзастосунку

3 Спеціальний розділ

3.1 Інструкція з розгортання вебзастосунку

3.2 Інструкція з наповнення вебзастосунку

3.3 Інструкція з популяризації та підтримки вебзастосунку

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

4.5 Обчислення накладних витрат

4.6 Складання кошторису витрат та визначення собівартості вебсайту

4.7 Розрахунок ціни вебсайту

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

5 Охорона праці, техніка безпеки та екологічні вимоги

5.1 Поняття праці та охорони праці

5.2 Дія електричного струму на організм людини

6 Висновки

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

5. Перелік графічного матеріалу:

1. Схема структурна вебсайту

2. Діаграма варіантів використання

3. ER-діаграма бази даних

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	Геннадій 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	11.06.2026	11.06.2026
12	Захист кваліфікаційної роботи	24.06.2026	24.06.2026

7. Дата видачі завдання: 05 березня 2026 р.

Студент

(підпис)

Іван БІДЛОВСЬКИЙ

Керівник роботи

(підпис)

Галина МАРЦІЯШ

ЗМІСТ

АНОТАЦІЯ	7
ANNOTATION	8
Вступ	9
1 Загальний розділ.....	10
1.1 Аналітичний огляд існуючих рішень.....	10
1.2 Технічне завдання	12
1.2.1 Найменування та область застосування	13
1.2.2 Призначення розробки.....	15
1.2.3 Вимоги до функціоналу вебзастосунку	17
1.2.4 Вимоги до програмної документації.....	17
1.2.5 Техніко-економічні показники	18
1.2.6 Стадії та етапи розробки	18
1.2.7 Порядок тестування та прийому.....	18
2 Розробка технічного та робочого проєкту.....	19
2.1 Розробка структури вебсервісу.....	19
2.2 Створення та верстка сторінок вебсервісу	20
2.3 Розробка структури бази даних	23
2.4 Програмування вебзастосунку.....	24
2.4.1 Написання клієнтської частини.....	26
2.4.2 Написання серверної частини.....	28
2.5 Тестування вебзастосунку.....	31
3 Спеціальний розділ	46
3.1 Інструкція з розгортання вебзастосунку.....	46
3.2 Інструкція з наповнення вебзастосунку	47
3.3 Інструкція з популяризації та підтримки вебзастосунку	49

					2026.KBP.122.421.1.00.00 ПЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бідловський І.			Літ.	Арк.	Аркуші
Перевір.		Марціаш Г.Я.				5	76
Реценз.					Розробка вебсервісу «Interiorix» Пояснювальна записка		
Н. Контр.		Приймак В.А.					
Затверд.							
					ВСП ТФК ТНТУ КН-421 м. Тернопіль		

4. ЕКОНОМІЧНИЙ РОЗДІЛ	51
4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР	51
4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи .	52
4.3. Розрахунок витрат на електроенергію	54
4.4. Розрахунок суми амортизаційних відрахувань розробки вебсервісу «Interiorix»	55
4.5. Обчислення накладних витрат	55
4.6. Складання кошторису витрат та визначення собівартості розробки вебсервісу «Interiorix»	56
4.7. Розрахунок ціни розробки вебсервісу «Interiorix»	56
4.8. Визначення економічної ефективності і терміну окупності капітальних вкладень	57
5 Охорона праці, техніка безпеки та екологічні вимоги	59
5.1 Поняття праці та охорони праці	59
5.2 Дія електричного струму на організм людини	60
Висновки.....	63
Перелік посилань	64
Додатки	65

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

АНОТАЦІЯ

Тема кваліфікаційної роботи: Розробка вебзастосунку для створення та реалізації індивідуального інтер'єру та екстер'єру «Interiorix».

Метою кваліфікаційної роботи є розробка вебсервісу для управління замовленнями у сфері створення та реалізації індивідуального інтер'єру та екстер'єру «Interiorix».

Пояснювальна записка складається з 5 розділів.

У загальній частині описуються аналітичний огляд існуючих рішень та аналіз технічного завдання розробки вебсервісу для управління замовленнями у сфері створення та реалізації індивідуального інтер'єру та екстер'єру «Interiorix»

У другому розділі представлено процес створення програмного продукту, опис та обґрунтування вибору структури та методу організації вхідних та вихідних даних, опис алгоритмів, інформаційних зв'язків, зовнішнє проектування програми, тестування та налагодження програм вебсервісу для управління замовленнями у сфері створення та реалізації індивідуального інтер'єру та екстер'єру «Interiorix».

В спеціальній частині описані процес інсталяції програмного продукту, інструкція з використання тестових наборів та інструкція з користування програмою «Interiorix».

Розрахунок вартості розробки та економічної ефективності приведено в економічній частині вебсервісу для управління замовленнями у сфері створення та реалізації індивідуального інтер'єру та екстер'єру «Interiorix».

Основні питання охорони праці та техніки безпеки розглянуто в п'ятому розділі.

Обсяг пояснювальної записки 76 сторінок.

До складу кваліфікаційної роботи входить графічна частина, що містить схему структурну вебсайту, діаграму варіантів використання, ER-діаграму бази даних і таблицю техніко-економічних показників, виконані на окремих аркушах формату А1.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ANNOTATION

The qualification thesis presents the development of the web application «Interiorix» for the creation and implementation of individual interior and exterior design. The aim of the work is to automate the receipt, processing and support of orders and to simplify interaction between clients and designers.

The explanatory note consists of five sections.

The general section provides an analytical review of existing solutions and an analysis of the technical requirements for the application.

The second section presents the process of creating the software product: the choice and justification of the structure and the organisation of input and output data, the description of algorithms and information links, the external design of the program, and the testing and debugging of the application.

The special section describes the installation of the software product, the use of test datasets and the user manual for the «Interiorix» application.

The economic section provides the cost calculation and the assessment of the economic efficiency of the development.

Occupational health and safety issues are considered in the fifth section.

The explanatory note comprises 76 pages.

The thesis includes a graphic part consisting of the website structural diagram, the use-case diagram, the database ER diagram, and the table of technical and economic indicators, presented on separate A1-format sheets.

					2026.KBP.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

У сучасному цифровому суспільстві розвиток інформаційних технологій стрімко охоплює всі сфери людської діяльності, зокрема й дизайну інтер'єру та екстер'єру. Автоматизація процесів, пов'язаних із прийомом, обробкою та супроводом замовлень, стає важливим чинником ефективної роботи компанії, що займається створенням індивідуальних дизайнерських рішень. З огляду на високий рівень конкуренції та зростаючі очікування клієнтів, виникає потреба у створенні зручного, функціонального й надійного вебервісу, який забезпечить якісну взаємодію між користувачем і дизайнерами.

Кваліфікаційну роботу присвячено розробці вебсервісу «Interiorix» – інноваційного сервісу для управління замовлень у сфері створення та реалізації індивідуального інтер'єру та екстер'єру. Його основне призначення оптимізувати процеси оформлення та супроводу замовлень, автоматизація обліку послуг і проєктів, спрощення комунікації між клієнтам та дизайнерами, а також забезпечені зручного контролю з виконанням робіт..

Під час реалізації проєкту використовуються сучасні технології: на стороні серверної частини – TypeScript з використанням фреймворку Nest.js і СУБД MongoDB, а на стороні клієнта – мова TypeScript та бібліотека React. Для полегшення розгортання використовується контейнеризацію за допомогою Docker, що забезпечує простоту масштабування та підтримки.

Проєкт має на меті продемонструвати практичні навички у сфері повного циклу розробки вебзастосунків: від аналізу вимог, проєктування бази даних і архітектури системи до реалізації функціоналу, тестування та підготовки до впровадження. У ході виконання роботи досліджено існуючі аналоги, визначено їх переваги та недоліки, а також сформовано концепцію, яка враховує актуальні потреби ринку.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

Для ефективної розробки вебсервісу для управління замовленнями у сфері створення та реалізації індивідуального інтер'єру та екстер'єру «Interiorix» необхідно провести детальний аналіз конкурентного середовища, зокрема вивчити функціональні можливості, переваги та недоліки провідних гравців на ринку реалізації індивідуального інтер'єру та екстер'єру. Такий підхід дозволяє виявити актуальні тренди, проблеми, які залишаються невирішеними, та сформувати унікальні конкурентні переваги майбутнього сервісу. На українському ринку найбільш популярними платформами є Houzz, Planner 5D та Homestyler, кожна з яких має свої особливості та цільову аудиторію.

1. Houzz

Є однією з найвідоміших міжнародних платформ у сфері дизайну інтер'єру, ремонту та облаштування житла. Сервіс поєднує каталог дизайнерських рішень, маркетплейс товарів і можливість пошуку професійних дизайнерів та підрядників.

Переваги:

- широкий каталог дизайнерських рішень та прикладів оформлення інтер'єру та екстер'єру ;д
- можливість взаємодії між клієнтами та професійними дизайнерами;
- сучасний та зручний інтерфейс із персоналізованими рекомендаціями.

Недоліки:

- значна частина функціоналу орієнтована на міжнародний ринок, що ускладнює адаптацію для локальних користувачів;
- обмежені можливості управління індивідуальними замовленнями в межах єдиної системи;
- деякі професійні функції доступні лише у платній версії .

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

2. Planner 5D

Цей онлайн- сервіс для створення дизайну інтер'єру та планування приміщень у 2D та 3D форматах. Платформа орієнтована як на професіоналів, так і на звичайних користувачів.

Переваги:

- можливість візуалізації дизайнерських рішень у 3D;
- простий інтерфейс для створення планів приміщення.

Недоліки:

- сервіс зосереджений переважно на візуалізацію, а не на повному циклі управління замовленнями;
- відсутність розвиненої системи взаємодії між клієнтами та виконавцями.

3. Homestyler

Homestyler є популярною платформою для моделювання інтер'єрів, яка дозволяє створювати дизайн-проекти з використанням готових шаблонів і 3D-візуалізації.

Переваги:

- сучасний інтерфейс і доступність для широкого кола користувачів;
- підтримка швидкого створення дизайн-проектів.

Недоліки:

- обмежена функціональність щодо автоматизації бізнес-процесів;
- відсутність комплексної системи управління клієнтським замовленнями.

У цьому контексті веб-сервіс Interiorix буде розроблений із врахуванням недоліків існуючих платформ, пропонуючи сучасний інтерфейс, автоматизацію основних операцій, інтеграцію з базами даних та високий рівень персоналізації для дизайнерів і користувачів.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

1.2 Технічне завдання

Метою створення вебсервісу «Interiorix» є розробка функціонального, безпечного та масштабованого онлайн-сервісу для взаємодії між дизайнерами та користувачами, що зацікавлені в створенні індивідуального інтер'єру або екстер'єру. Застосунок повинен об'єднувати у собі переваги сучасного інтерфейсу, гнучкого пошуку, широких можливостей керування замовленнями та інтегрованої комунікації. Сервіс включатиме публічну частину для всіх відвідувачів, авторизовані можливості для зареєстрованих користувачів (клієнтів та дизайнерів) та панель адміністратора для керування вмістом, модерації контенту і контролю за активністю на платформі.

Застосунок буде реалізовано як клієнт-серверна архітектура. Серверна частина розробляється з використанням мови програмування TypeScript та фреймворку Nest.js, який забезпечує ефективну побудову RESTful API, підвищену безпеку, розширюваність та підтримку модульного підходу. Усі структуровані дані (користувачі, послуги, повідомлення, відгуки, статистика) зберігатимуться в нереляційній базі даних MongoDB, яка забезпечує високу продуктивність, підтримку транзакцій та гнучкість при виконанні складних запитів. Для забезпечення надійної роботи з клієнтом застосунок матиме користувацький інтерфейс, реалізований за допомогою бібліотеки React з мовою TypeScript, що дозволить створити сучасний, адаптивний і зручний інтерфейс користувача з високим рівнем інтерактивності.

Серед основних функціональних можливостей для кінцевого користувача: реєстрація, авторизація, редагування особистого профілю, надсилання заявок, переглядів статусу заявок, використання розширеного пошуку за такими критеріями як тип, площа, ціна, стан об'єкта, тощо. Кожен проєкт буде представлений у вигляді картки з фото, коротким описом.

Для адміністратора буде реалізовано особистий кабінет із можливістю додавання нових проєктів, редагування існуючих, перегляду статистики за переглядами, управління запитами від користувачів та керування контактними

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

даними. Важливою особливістю буде автоматична перевірка полів без можливість публікацій не валідних даних.

Панель адміністратора надаватиме доступ до всіх проєктів і користувачів у системі. Адміністратор зможе підтверджувати або відхиляти заявки, змінювати їх статус, переглядати аналітику за днями, місяцями, відстежувати активність, а також змінювати інформацію уже про наявні проєкти. Для взаємодії між користувачем і агентом буде відбуватися як поза межами застосунку так і всередині за допомогою системних повідомлень.

Система автентифікації передбачатиме реєстрацію за email та паролем, верифікацію пошти. Для безпеки даних усі паролі зберігатимуться у хешованому вигляді, а REST API буде захищено через JWT-токени. Доступ до адміністративних функцій обмежується ролями, які зберігаються в базі даних.

Фронтенд-застосунок працюватиме як повноцінний вебсервіс, що дозволить користувачам виконувати всі дії в одній місці. React-компоненти будуть побудовані за модульною структурою з повторно використовуваними блоками (карти, фільтри, форми), а для стилізації буде використано бібліотеки Tailwind CSS. Весь клієнтський код буде оптимізовано для роботи з комп'ютером. Також буде реалізовано lazy loading для зображень та компонентів, які не потребують негайного рендерингу.

Таким чином, технічне завдання охоплює повний цикл розробки функціонального, захищеного та масштабованого сервісу з багаторівневою роллю доступу, інтерактивним інтерфейсом та інтегрованою логікою для потреб створення індивідуального інтер'єру та екстер'єру.

1.2.1 Найменування та область застосування

Кваліфікаційна робота на тему «Розробка вебсервісу для управління замовленнями у сфері створення та реалізації індивідуального інтер'єру та екстер'єру «Interiorix» і є інформаційно-функціональною платформою, що призначена для автоматизації процесів, пов'язаних з операціями у сфері дизайну

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

інтер'єру та екстер'єру. Сервіс «Interiorix» покликаний забезпечити сучасне середовище для ефективної взаємодії між двома основними сторонами: дизайнерами та кінцевими споживачами послуг (клієнтами). Платформа дозволяє користувачам вибирати готові рішення для створення дизайну інтер'єру та екстер'єру, знаходити відповідні рішення за допомогою фільтрів, створювати заявку, а також керувати ними через особистий кабінет.

Область застосування вебсервісу охоплює як користувачів, які шукають готові рішення, так і саму дизайнерську агенцію. Окрему групу користувачів складають адміністратори платформи, які здійснюють модерацію контенту, аналітичний контроль, верифікацію користувачів та загальне керування функціонуванням системи.

Застосунок побудовано за клієнт-серверною архітектурою та функціонує у вигляді вебсервісу, доступного з будь-якого пристрою з браузером. Система не потребує встановлення додаткового програмного забезпечення на пристрої користувача, що значно розширює її доступність і зручність використання. Усі дані зберігаються у захищеній базі даних, доступ до якої контролюється відповідно до ролей користувачів – звичайний відвідувач, адміністратор, менеджер, дизайнер. Це дозволяє забезпечити високий рівень безпеки, персоналізації та гнучкого керування ресурсами платформи.

Окрему увагу в застосунку приділено UX/UI дизайну, який адаптований до потреб різних категорій користувачів, включаючи людей з базовим рівнем цифрової грамотності. Інтерфейс «Interiorix» реалізований таким чином, щоб мінімізувати кількість дій для досягнення цілей користувача (наприклад, пошук рішення, створення заявки, отримання сповіщень).

У результаті застосунок «Interiorix» позиціонується як універсальний веб-сервіс, що поєднує інтереси як приватних осіб, так і бізнес-сегменту, підвищуючи ефективність, прозорість і цифрову зрілість ринку дизайну готових рішень. Завдяки поєднанню технологічної надійності, сучасного дизайну та багатофункціональності, він може успішно впроваджуватись у якості комерційного продукту, інфраструктурного рішення або науково-освітнього кейсу.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

1.2.2 Призначення розробки

Призначенням розробки вебсервісу «Interiorix» є автоматизація діяльності дизайнерського агентства через створення інноваційного цифрового інструменту, який забезпечить ефективне управління проектами, прозору взаємодію між дизайнерами, менеджерами та клієнтами, оптимізацію бізнес-процесів та покращення якості обслуговування. У сучасних умовах стрімкого розвитку технологій, а також зростання вимог клієнтів до швидкості й зручності отримання послуг, особливо актуальним стає впровадження інтелектуальних програмних рішень у сферу дизайну. Вебсервіс «Interiorix» покликаний не лише замінити застарілі паперові або фрагментовані електронні засоби обліку, а й створити єдину екосистему для організації, презентації, пошуку та супроводу угод з дизайну індивідуальних рішень інтер'єру та екстер'єру.

Одна з головних функцій вебсервісу – облік кейсів та заявок на створення дизайнерських рішень, що передбачає створення централізованої бази даних, в якій кожен проєкт має детальний опис: назва, площа, кількість кімнат, поверховість, адреса, стан, ціна, фотографії. Це дозволяє дизайнерам та адміністраторам оперативно додавати нові об'єкти, редагувати наявні пропозиції та контролювати актуальність даних. Наявність повної та достовірної інформації є ключовим фактором довіри клієнтів до агентства.

Застосунок забезпечить зручний і гнучкий інструмент пошуку та фільтрації об'єктів за основними параметрами, такими як вартість, матеріали, фасадом, тощо. Це дозволяє клієнтам швидко знайти найбільш релевантні варіанти, що відповідають їхнім індивідуальним критеріям. Завдяки цьому, користувачі отримують зручний інтерфейс для вибору оптимального житлового або комерційної рішення у бажаному дизайні.

Додатковою перевагою є можливість залишення заявок на виконання послуг. Користувачі можуть в один клік відправити запит на створення дизайну у бажаному стилі. Усі запити фіксуються в системі та мають статус (наприклад, "в обробці", "підтверджено", "відхилено"), що дозволяє прозоро відслідковувати

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

історію взаємодії. Для дизайнерів реалізовано сповіщення про нові заявки з можливістю перегляду деталей і швидкої відповіді клієнту, що пришвидшує процес комунікації.

Кабінет користувача є важливою складовою персоналізації сервісу. Залежно від ролі, користувач отримує різний набір функціоналу. Для клієнта передбачено можливість переглядати статуси заявок, слідкувати за історію їх виконання та керування особистими даними. Для дизайнерів існують інструменти додавання нових проєктів у портфоліо, перегляду заявок, оновлення статусів, створення статистичних звітів. Менеджери проєктів отримують повний доступ до керування проєктами, перегляду ключових характеристик та зміну самого проєкту. Адміністратори системи мають розширений доступ для керування користувачами, контентом і аналітикою.

Важливим є те, що вебсервіс дозволяє покращити внутрішню ефективність дизайнерського агентства. Завдяки централізованому обліку, автоматизованим процесам та аналітичним панелям, керівництво агентства може оперативно приймати рішення, контролювати виконання планів, аналізувати активність працівників та виявляти вузькі місця в обслуговуванні клієнтів. Усі дані, зібрані в межах платформи, можуть використовуватись для бізнес-аналітики, прогнозування попиту, розробки маркетингових кампаній тощо.

Розробка вебсервісу «Interiorix» орієнтована на сучасні технології. Серверна частина реалізується з використанням TypeScript та фреймворку Nest.js, що забезпечує надійність, безпеку та масштабованість. У якості бази даних використовується MongoDB — потужна нереляційна СУБД. Фронтенд частина створюється на основі React з TypeScript, що дозволяє розробити інтуїтивно зрозумілий інтерфейс, гнучкий до змін.

Платформа розробляється з урахуванням принципів безпеки, зокрема реалізовано автентифікацію користувачів, контроль доступу до даних, захист від SQL-ін'єкцій та XSS-атак.

Отже, розробка вебсервісу «Interiorix» спрямована на підвищення ефективності дизайнерського агенства шляхом автоматизації його діяльності,

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

зниження навантаження на персонал, покращення комунікації з клієнтами та формування позитивного іміджу інноваційного агентства. У поєднанні з сучасними вебтехнологіями та фокусом на зручність користувача, «Interiorix» є перспективним рішенням для впровадження у практичну діяльність ринку готових дизайнерських рішень України.

1.2.3 Вимоги до функціоналу вебзастосунку

Для забезпечення ефективної роботи вебсервісу «Interiorix» він повинен включати таку функціональність:

- збереження інформації про як готові проекти з портфоліо компанії, так і про ті, які створюються з побажання клієнта з деталями, такими як назва, матеріали, площа, ціна, статус, тощо;
- організація даних за категоріями;
- пошук та фільтрація об'єктів за заданими критеріями (наприклад, площа, ціна, тип);
- можливість клієнтів залишати заявки;
- модуль управління доступом із можливістю створення профілів для адміністраторів, дизайнерів, менеджерів і клієнтів;
- панель аналітики для відстеження статистики запитів і готовності проектів.

1.2.4 Вимоги до програмної документації

По закінченню розробки програмного забезпечення потрібно підготувати таку документацію:

- інструкція інсталяції програми;
- загальні відомості про можливості програми;
- інструкція з експлуатації.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

1.2.5 Техніко-економічні показники

Планові техніко-економічні показники розробки вебсервісу, визначені в економічному розділі, складають:

- собівартість розробки – 72 977 грн;
- планова ціна продукту – 113 844 грн;
- плановий прибуток від реалізації – 40 867 грн;
- чиста теперішня вартість (ЧТВ) – 28 654 грн.; термін окупності капітальних вкладень – 2,1 року.

1.2.6 Стадії та етапи розробки

Розробка вебсервісу «Interiorix» буде мати такі стадії:

- аналіз функціональних вимог;
- проектування архітектури вебсервісу;
- розробка серверної та клієнтської частин;
- тестування функціоналу та безпеки;
- написання документації;
- впровадження програми.

1.2.7 Порядок тестування та прийому

Для забезпечення якості вебсервісу важливо встановити наступний порядок контролю та прийому:

- контроль на кожному етапі розробки;
- перевірку відповідності технічному завданню, правильності архітектури, функціональності коду та зручності інтерфейсу;
- проведення перевірок у реальних умовах використання (фінальне тестування).

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури вебсервісу

Розробка структури вебсервісу «Interiorix» є важливим етапом, який визначає архітектурні рішення, логіку взаємодії між складовими частинами системи, а також ефективність, масштабованість і зручність її подальшого використання та супроводу. Структура вебзастосунку передбачає чіткий поділ на логічні рівні: клієнтський інтерфейс (frontend), серверну частину (backend) і рівень роботи з базою даних (data layer). Такий підхід дає змогу розмежувати відповідальність кожного модуля та забезпечити високу якість розробки, простоту тестування та гнучкість під час внесення змін.

Клієнтський інтерфейс розробляється з використанням сучасної JavaScript-бібліотеки React у поєднанні з мовою TypeScript. Це забезпечує безпеку, підвищує надійність коду і дозволяє реалізувати масштабовану компонентну архітектуру. Уся логіка користувацького інтерфейсу будується за принципом розподілу на багаторазові компоненти, які відповідають за окремі елементи інтерфейсу, наприклад: форми авторизації, фільтри пошуку, картки проєктів, модальні вікна, панель користувача тощо. Це дозволяє легко підтримувати й розвивати проєкт, а також зменшує повторення коду. Основна увага приділяється створенню інтуїтивно зрозумілого і зручного інтерфейсу. Користувачі матимуть змогу здійснювати пошук готових кейсів, переглядати детальну інформацію, залишати заявки, залишати відгуки, а також керувати своїм профілем.

Серверна частина вебзастосунку реалізується на основі фреймворку Nest.js, що використовується разом із мовою програмування TypeScript. Nest.js надає широкі можливості для побудови RESTful API, які використовуються для обміну даними між клієнтською частиною та сервером. Архітектура бекенду будується за принципами розділення на шари: контролери, сервіси, репозиторії, що забезпечує гнучку структуру проєкту. Контролери відповідають за прийом HTTP-запитів і повернення відповідей, сервісний рівень обробляє бізнес-логіку, а репозиторії

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

взаємодіють безпосередньо з базою даних. Таке розділення дозволяє легко модифікувати окремі частини системи без впливу на інші. Також реалізовано механізми автентифікації та авторизації користувачів за допомогою JSON Web Token (JWT), що забезпечує захист даних і обмеження доступу відповідно до ролей (адміністратор, агент, клієнт).

Значну увагу приділено розробці бази даних, яка є основою для зберігання всієї інформації, пов'язаної з проєктами, користувачами, заявками, повідомленнями, статусами проєктів та історією змін. У якості системи управління базами даних обрано MongoDB – надійну нереляційну СУБД. Структура бази даних проєктується з урахуванням нормалізації, цілісності даних та забезпечення швидкої роботи при масштабуванні. Використання зовнішніх ключів забезпечує зв'язки між таблицями, дозволяючи зберігати цілісність інформації.

Окрім основної структури, в системі реалізовано окремі модулі для обробки запитів на перегляд, система фільтрації та сортування даних, панель адміністратора з аналітикою та можливістю управління користувачами й об'єктами дизайну. У структурі фронтенду виділено логіку маршрутизації для переходу між сторінками (наприклад, головна сторінка, сторінка об'єкта, форма реєстрації, особистий кабінет), що дозволяє користувачу зручно переміщатися між розділами застосунку.

Таким чином, структура вебсервісу «Interiorix» побудована за принципами сучасної веб-розробки: розподілена архітектура, розмежування обов'язків, використання перевірених технологій, захищений обмін даними між клієнтом і сервером. Це дозволяє створити стабільний, надійний і зручний інструмент для автоматизації дизайнерського агентства, який легко розширюється й адаптується під потреби кінцевого користувача.

2.2 Створення та верстка сторінок вебсервісу

Процес створення та верстки сторінок вебсервісу «Interiorix» є важливим етапом реалізації клієнтської частини системи, оскільки саме інтерфейс взаємодії з користувачем формує перше враження про сервіс і визначає зручність

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

користування. Основною метою на цьому етапі є розробка зручного, інтуїтивно зрозумілого дизайну, що враховує сучасні тенденції у веброботці, потреби користувачів та специфіку ринку дизайнерських рішень. Створення інтерфейсу здійснювалося з використанням бібліотеки React у поєднанні з мовою програмування TypeScript та бібліотекою Tailwind CSS, яка дозволяє швидко реалізувати стилізовані компоненти з дотриманням принципів адаптивного дизайну.

Процес розробки інтерфейсу розпочинався з побудови прототипу дизайну, в якому були визначені основні сторінки застосунку, їх структура, ієрархія елементів, логіка навігації між ними. Основними сторінками вебзастосунку стали: головна сторінка, сторінка з переліком об'єктів, сторінка детального перегляду об'єкта, форма додавання нового об'єкта, сторінка подання заявки, особистий кабінет користувача, сторінка реєстрації/авторизації та адміністративна панель. Кожна сторінка реалізовувалась як окремий React-компонент з урахуванням повторного використання спільних елементів, таких як шапка (Header), футер (Footer), навігаційне меню, форма пошуку, картка проєкту тощо.

Головна сторінка має адаптивне розміщення контенту, включає банер із заклик до дії, блок популярних пропозицій та фільтр для швидкого переходу до пошуку готових дизайнерських рішень. Важливою частиною стала реалізація механізму фільтрації, що дозволяє користувачеві обрати тип об'єкта, діапазон цін, площу, кількість кімнат та інші параметри. Цей функціонал забезпечується за допомогою React-станів і передачі даних між компонентами, що дозволяє швидко змінювати результати без перезавантаження сторінки. Дані завантажуються через API-запити до серверної частини, що реалізована на Nest.js, а відповіді виводяться у вигляді карток проєктів у вигляді сітки.

На сторінці детального перегляду об'єкта представлена розширена інформація: фотографії, опис, характеристики, місцезнаходження на карті, контактна інформація. З технічної точки зору ця сторінка динамічно підвантажує інформацію з бази даних відповідно до ідентифікатора обраного об'єкта, що передається через параметри URL. Для зручності користувача

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

передбачено реалізацію галереї з фотографіями, прокручуваного опису та блоку з подібними об'єктами.

Особистий кабінет реалізований з урахуванням ролей користувача. У разі входу адміністратора відображається його головне меню, де представлені додаткові функції: можливість додавання/редагування об'єктів, перегляду активних заявок, управління користувачами. Для звичайного клієнта особистий кабінет включає можливість створення, статус угод. Вся інформація завантажується динамічно, а компоненти повторно використовуються у вигляді вкладок або секцій. Особливу увагу було приділено захисту персональних даних, реалізації перевірки авторизації при доступі до особистої інформації та забезпеченню безпеки форм через валідацію з обох боків – як на клієнті, так і на сервері.

Також була реалізована сторінка додавання тинової заявки, яка доступна лише авторизованим користувачам, дизайнерам та менеджерам. Ця форма включає обов'язкові поля: кейс з портфолію, тип послуги, коротка назва заявки. Для завантаження зображень реалізовано попередній перегляд (preview) з можливістю видалення до підтвердження збереження. Всі поля проходять перевірку на коректність введення даних, а при відправленні форми інформація зберігається в базу даних MongoDB.

Важливою складовою став процес тестування і перевірки працездатності кожної сторінки. Усі компоненти проходили тестування, проводилося ручне тестування верстки, перевірка кросбраузерної сумісності та швидкодії. На завершальному етапі кожна сторінка була інтегрована в загальну систему маршрутизації за допомогою React Router, що забезпечило логічну й зручну навігацію між усіма екранами застосунку.

Таким чином, створення та верстка сторінок вебсервісу «Interiorix» реалізовано відповідно до сучасних принципів розробки вебінтерфейсів. Результатом є інтуїтивно зрозумілий, динамічний і адаптивний інтерфейс, який задовольняє потреби як пересічних користувачів, так і працівників дизайнерського агентства. Це забезпечує високий рівень зручності, функціональності та довіри до

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

сервісу.

2.3 Розробка структури бази даних

Для зберігання та обробки інформації у вебсервісу «Interiorix» розроблена реляційна структура бази даних, яка забезпечує ефективне управління об'єктами дизайну, користувачами, замовленнями, відгуками та звітністю. Взаємозв'язки між таблицями реалізовано за допомогою зовнішніх ключів, що гарантує цілісність даних. Основу СКБД становить MongoDB. Нижче наведено детальний опис основних колекцій бази даних.

- 1) «users» містить основну інформацію про користувачів платформи;
- 2) «roles» містить ролі користувачів і права доступу;
- 3) «clientprofiles» зберігає додаткову інформацію про клієнтів;
- 4) «staffprofiles» містить дані про працівників, дизайнерів та менеджерів;
- 5) «orders» використовується для зберігання заявок і замовлень клієнтів;
- 6) «projects» містить інформацію про дизайнерські проекти, їх статус, бюджет, терміни та відповідальних осіб;
- 7) «materials» зберігає інформацію про матеріали, їх ціну, кількість на складі та постачальників;
- 8) «reviews» зберігає відгуки клієнтів;

Структура бази даних забезпечує логічне моделювання всіх бізнес-процесів, пов'язаних з об'єктами нерухомості, орендою, замовленнями, переглядами та роботою користувачів. Всі таблиці логічно пов'язані між собою через зовнішні ключі, що дозволяє ефективно виконувати складні запити, забезпечує цілісність даних та можливість розширення системи. Такий підхід дає змогу реалізувати як базовий функціонал (пошук, запити, облік об'єктів), так і додаткові можливості (аналітика, звітність, відгуки) у рамках архітектури, що легко масштабується.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

2.4 Програмування вебзастосунку

Процес програмування вебсервісу є одним з ключових етапів у реалізації функціональності системи управління нерухомістю. Основна мета програмування – реалізувати усі логічні компоненти, інтегрувати базу даних, реалізувати зручний і сучасний інтерфейс користувача, а також забезпечити надійну комунікацію між клієнтською та серверною частинами. Для досягнення поставлених цілей було обрано стек технологій, який поєднує TypeScript, фреймворк Nest.js на бекенді, MongoDB як систему керування базами даних, та TypeScript з бібліотекою React для фронтенд-розробки.

Розробка бекенд-частини розпочалася з основних класів програми. Було створено основну структуру проєкту з використанням шаблону MVC (Model-View-Controller), де моделі відповідають сутностям бази даних (квартири, замовлення, користувачі, житлові комплекси тощо), контролери реалізують точки входу REST API, а сервіси займаються обробкою бізнес-логіки. Кожна сутність бази даних має відповідний JPA-репозиторій, який дозволяє ефективно виконувати CRUD-операції, фільтрацію, сортування та пошук без необхідності ручного написання SQL-запитів. Було реалізовано мапінг між сутностями та DTO-об'єктами, що дозволило розділити внутрішню структуру даних від зовнішніх API-структур, забезпечуючи тим самим безпеку та гнучкість розробки.

Особливу увагу приділено аутентифікації та авторизації користувачів. У застосунку реалізовано JWT (JSON Web Token) механізм для безпечної передачі токенів доступу між клієнтом і сервером. Завдяки ролям (USER, ADMIN, DESIGNER, MANAGER) реалізована рольова модель доступу до ресурсів, що забезпечує контрольовану взаємодію користувачів з системою залежно від їхніх прав. Наприклад, лише адміністратор може створювати або редагувати інформацію про здані проєкти, тоді як звичайний користувач має доступ лише до пошуку та перегляду їх

Бекенд також включає реалізацію API для обробки записів на перегляд об'єктів, створення заявок, ведення звітів за замовленнями і переглядами,

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

бронювання квартир, управління улюбленими об'єктами тощо. Запити реалізовані згідно з REST-принципами, використовуються HTTP-методи GET, POST, PUT, DELETE, що відповідають відповідно операціям читання, створення, оновлення та видалення ресурсів..

На стороні бази даних MongoDB створено всі необхідні документи згідно зі структурою, розробленою на етапі проектування. Створені відповідні зв'язки між таблицями (foreign keys), індекси для пришвидшення пошуку та фільтрації, налаштовано каскадне оновлення та видалення. Використання MongoDB обумовлено її стабільністю, потужними інструментами для роботи з транзакціями та широкою підтримкою розширених типів даних.

Фронтенд-частина розроблена з використанням бібліотеки React та мови TypeScript, що дозволяє створювати типізовані, зручні у підтримці та надійні компоненти інтерфейсу. Для організації стану додатку використовується Context API та локальні хук-стани (useState, useEffect). Компоненти поділені на повторно використовувані (наприклад, картка квартири, форма авторизації, модальне вікно перегляду), сторінкові компоненти (наприклад, сторінка каталогу, детальна сторінка об'єкта) та службові компоненти (наприклад, маршрутизація, навігація, хедер/футер). Для маршрутизації використано бібліотеку React Router v6, яка дозволяє налаштовувати маршрути відповідно до ролі користувача та забезпечує захист приватних маршрутів.

Усі дані, які відображаються на інтерфейсі, надходять з бекенду через REST API у форматі JSON. Для асинхронних запитів до API використано бібліотеку axios, яка дозволяє налаштовувати заголовки, токени авторизації та обробляти помилки запитів. Використано бібліотеки стилізації, такі як Tailwind CSS, які забезпечують сучасний вигляд елементів та підвищують зручність використання вебінтерфейсу.

Окремо реалізовані інтерфейси кабінетів для різних типів користувачів. Наприклад, клієнт бачить свої заявки, відгуки та доступні об'єкти, може залишити заявку на створення схожого рішення. Дизайнер відслідковує свої проекти, їх статуси та має можливість додавати свої власні. Менеджер керує усім процесом, переглядає доступні матеріали, керує проектами. Адміністратор має повний

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

контроль над всіма розділами сайту, включаючи, об'єкти, користувачів та модерацію контенту.

Розробка велась у локальному середовищі з використанням системи контролю версій Git. Код розподілено між окремими репозиторіями для фронтенду та бекенду, що полегшує підтримку та розгортання системи. Для тестування бекенд API застосовувалися інструменти Postman та Swagger UI, що дозволило перевірити всі сценарії взаємодії з системою.

Таким чином, програмування вебсервіс включало повний цикл реалізації логіки на бекенді та фронтенді, з урахуванням сучасних стандартів безпеки, масштабованості, зручності використання та модульності.

В результаті було створено надійну архітектуру, яка дозволяє ефективно працювати з великим обсягом даних, забезпечувати зручну взаємодію користувачів з дизайнерськими рішеннями та підтримувати розширення функціональності в майбутньому.

2.4.1 Написання клієнтської частини

Клієнтська частина вебсервісу «Interiorix» є важливою складовою всієї системи, оскільки саме через неї здійснюється безпосередня взаємодія користувача з функціоналом сервісу. Основна мета клієнтської частини полягає у створенні зручного, інтуїтивно зрозумілого, швидкого й адаптивного інтерфейсу, що дозволяє користувачам переглядати нерухомість, фільтрувати пропозиції, реєструватися, авторизовуватися, здійснювати замовлення, додавати пергядати статус об'єктів, керувати заяками, тощо.

Для реалізації клієнтської частини було обрано сучасний технологічний стек – TypeScript у поєднанні з бібліотекою React. Використання бібліотеки React дозволяє будувати масштабовані інтерфейси з повторно використовуваних компонентів, що спрощує подальше розширення застосунку та полегшує його підтримку. TypeScript, у свою чергу, забезпечує статичну типізацію, що дозволяє уникати багатьох помилок на етапі розробки та підвищує стабільність коду.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

Проект було реалізовано з використанням компонентно-орієнтованого підходу: кожен елемент інтерфейсу – форма, список об'єктів, фільтр, сторінка перегляду тощо – представлений окремим функціональним компонентом. Для керування станом застосунку використано React Hooks (useState, useEffect, useContext) у поєднанні з контекстним API, що дозволяє централізовано зберігати дані про авторизованого користувача, обрані об'єкти, активні фільтри та інші глобальні стани.

Клієнтська частина активно взаємодіє з бекендом через REST API. Взаємодія реалізована за допомогою бібліотеки axios, яка надає зручний інтерфейс для виконання HTTP-запитів. Кожен компонент, який потребує отримання або відправлення даних, викликає відповідну функцію сервісу, що інкапсулює запити до серверу та обробку помилок.

Після запуску вебсервіс завантажує головну сторінку зі списком актуальних проектами.. Кожна картка включає ключову інформацію – фото, назву, – а також кнопки для оформлення замовлення.

Окрема увага приділялася розробці сторінки перегляду конкретного проекту. На ній виводиться повна інформація про проект, а також можливість. Якщо користувач не авторизований – він буде перенаправлений на сторінку входу (/login), що реалізується через перевірку контексту авторизації та механізм захищених маршрутів (ProtectedRoute).

Для сторінок реєстрації та авторизації реалізовано валідацію на клієнтському рівні (через регулярні вирази та обробку форм), а також зворотний зв'язок при помилках (виведення повідомлень сервера у вигляді alert або snackbar). Після авторизації токен зберігається в localStorage, а всі запити до захищених ресурсів автоматично містять заголовок Authorization: Bearer <token>.

Користувач має можливість переглянути свої заявки у власному кабінеті. Ця сторінка відображає замовлення (/orders), заявки, активні угоди, а також історію усіх проектів. Якщо користувач є адміністратором, йому доступна можливість керувати об'єктами (додавати, редагувати, видаляти), переглядати звіти та змінювати статуси запитів.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Візуальне оформлення клієнтської частини реалізовано за допомогою Tailwind CSS, що забезпечує адаптивний, сучасний та уніфікований інтерфейс із використанням утилітарних CSS-класів. Сторінки стилізовані з урахуванням принципів доступності та зрозумілості.

Маршрутизація у клієнтському застосунку організована за допомогою react-router-dom. Всі основні сторінки мають окремі компоненти, що відповідають за логіку та візуальне представлення. У випадку переходу на неіснуючий маршрут відображається сторінка 404.

Особлива увага приділялася оптимізації продуктивності. Для обмеження кількості HTTP-запитів використовуються кешовані відповіді, мемоізація компонентів (React.memo) та оптимізація ререндерів. Після авторизації користувача, застосунок одноразово отримує усі необхідні дані та зберігає їх у глобальному стані, що дозволяє уникати зайвих викликів до API при навігації.

Таким чином, клієнтська частина вебсервісу реалізує всі основні функції взаємодії з користувачем та забезпечує логіку для роботи із сервісом. Її реалізація ґрунтується на сучасних принципах компонентного підходу, реактивності, повторного використання логіки та модульності. В результаті створено надійний, масштабований і зручний фронтенд, що дозволяє якісно та ефективно вирішувати задачі користувачів агентства нерухомості.

2.4.2 Написання серверної частини

Серверна частина вебзастосунку «Interiorix» є ключовим елементом, що забезпечує логіку обробки даних, зберігання інформації у базі даних та взаємодію з клієнтською частиною через API. Основними завданнями серверної частини є: обробка запитів від користувачів, валідація вхідних даних, взаємодія з базою даних через об'єктно-реляційне відображення (ORM), реалізація бізнес-логіки та контроль доступу до ресурсів.

Для побудови серверної частини було обрано стек технологій на основі TypeScript та Nest.js. Цей фреймворк дозволяє створювати RESTful-сервіси з

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

чіткою структурою, високою продуктивністю та гнучкими можливостями конфігурації. Доступ до бази даних здійснюється за допомогою MongoDB.

У системі передбачено такі основні доменні сутності, які відповідають реальним об'єктам:

a) Сутності (entity):

- User;
- Role;
- Clientprofile;
- Staffprofile;
- Order;
- Project;
- Material;
- Review.

Розглянемо детальніше класи моделі програми:

1) User:

містить основну інформацію про користувачів платформи;

2) Role:

- містить ролі користувачів і права доступу;

3) Clientprofile:

- зберігає додаткову інформацію про клієнтів;

4) Staffprofile:

- містить дані про працівників, дизайнерів та менеджерів;

5) Order:

- використовується для зберігання заявок і замовлень клієнтів;

6) Project:

– містить інформацію про дизайнерські проекти, їх статус, бюджет, терміни та відповідальних осіб;

7) Material:

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

– зберігає інформацію про матеріали, їх ціну, кількість на складі та постачальників;

8) Review:

– зберігає відгуки клієнтів.

Рівень доступу до даних у вебсервері реалізується за допомогою сервісів. Використано Mongoose, який дозволяє легко взаємодіяти з базою даних за допомогою інтерфейсів та анотацій. Ці сервіси допомагають взаємодіяти з базою даних, ізолюючи бізнес-логіку від прямих взаємодій з джерелом даних.

При проектуванні прикладного інтерфейсу сервера «Interiorix» використовуються принципи RESTful архітектури. Всі ендпоінти базуються на стандартних HTTP-методах: GET, POST, PUT, DELETE.

Нижче наведено опис основних ендпоінтів.

1) Авторизація:

– POST api/auth/login;

2) Перехід у особистий кабінет:

– GET api/auth/me;

3) Завантаження файлів проєкту:

– POST api/files/upload;

4) Отримання конкретних файлів:

– GET api/files/{id};

5) Отримання інформації на головній сторінці особистого кабінету:

– GET api/dashboard/summary;

6) Перегляд усіх проєктів:

– GET api/projects.;

7) Створення нового проєкту:

– POST api/projects;

8) Отримання усіх власних проєктів:

– GET api/projects/mine;

9) Отримання певного проєкту за його ідентифікатором:

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

- GET api/projects/{id};
- 10) Отримання даних про оплату:
 - GET api/payment;
- 11) Перегляд чеків:
 - GET api/receipts;
- 12) Аналіз доступних матеріалів:
 - GET api/materials;
- 13) Отримання даних про усіх користувачів:
 - GET api/users;
- 14) Створення нового постачальника матеріалів:
 - POST api/suppliers;
- 15) Перегляд доступних постачальника матеріалів:
 - GET api/suppliers;
- 16) Видалення конкретного постачальника:
 - DELETE api/suppliers/{id};

Написання серверної частини вебсервісу «Interiorix» базувалося на принципах чистої архітектури, розділення обов'язків та сучасних підходах до реалізації RESTful-сервісів. Доменна модель відображає реальну структуру бізнесу агентства нерухомості, а API дозволяє зручно і безпечно взаємодіяти з даними. Реалізація контролю доступу, валідації даних, ролей користувачів та звітності дозволяє застосунку бути не лише інформативним, а й функціонально повноцінним інструментом для автоматизації роботи агенції.

2.5 Тестування вебзастосунку

При вході на сайт нас зустрічає головний екран. Результат представлено на рисунку 2.1.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

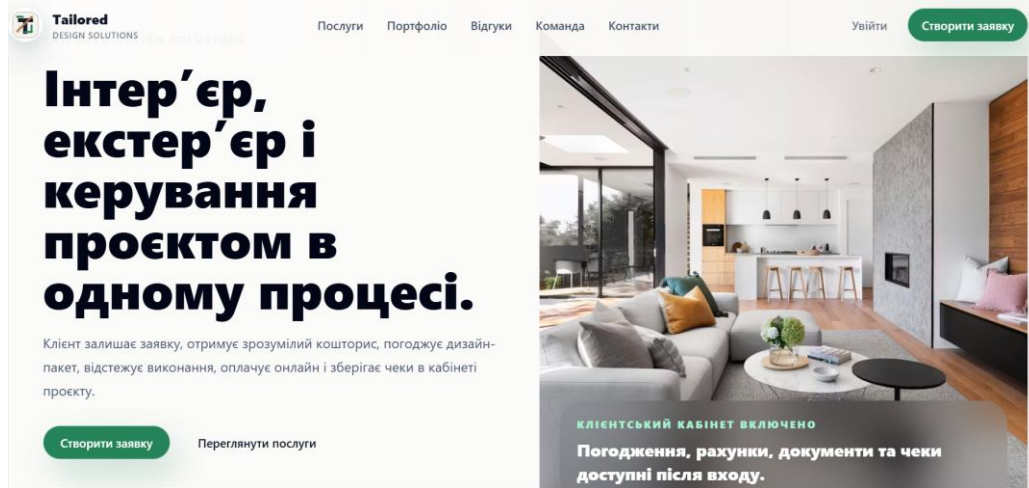


Рисунок 2.1 – Головний екран

Популярні проєкти наведено на рисунку 2.2.

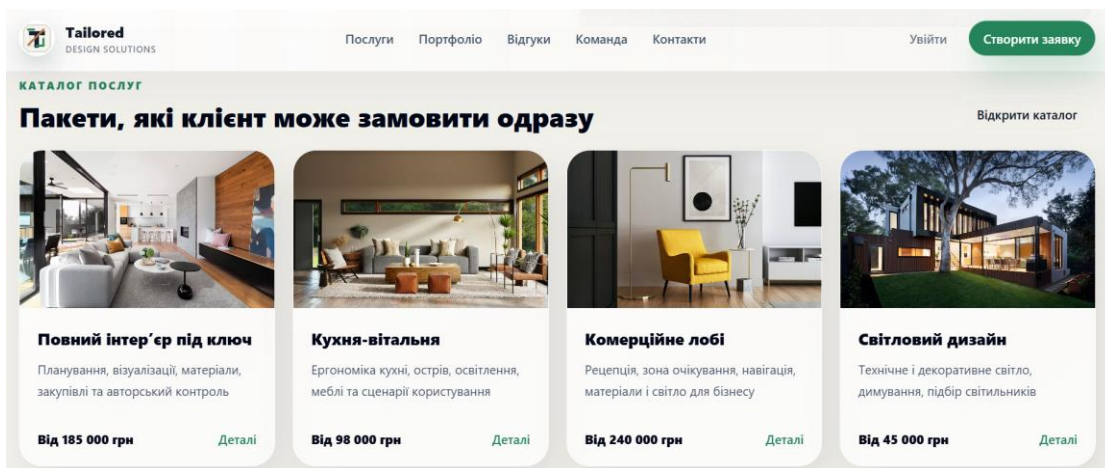


Рисунок 2.2 – Популярні проєкти

Фрагменти портфолію та декілька відгуків представлено на рисунку 2.3.

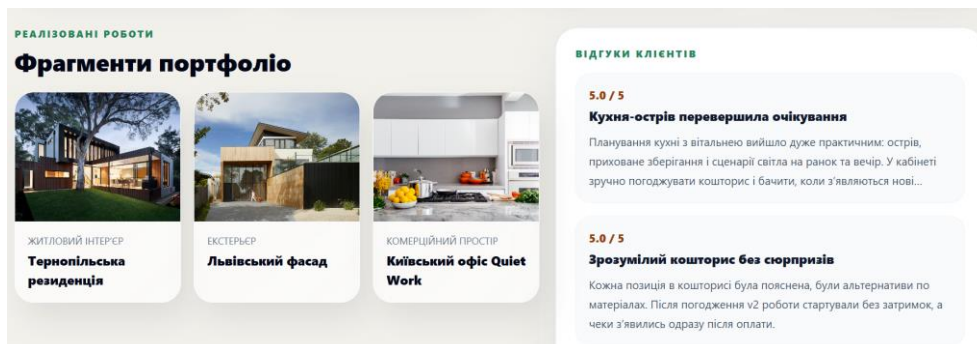


Рисунок 2.3 – Фрагменти портфолію та відгуки

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

Панель «Портфоліо», для перегляду доступних готових варіантів представлено на рисунку 2.4.

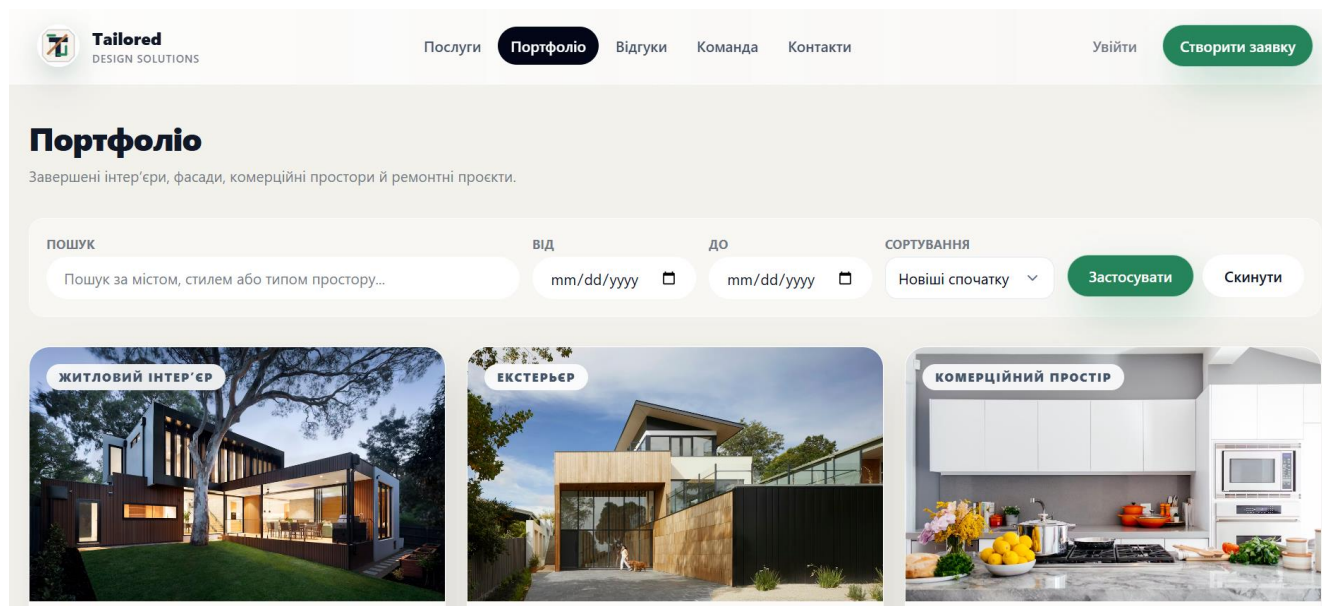


Рисунок 2.4 – Панель «Портфоліо»

Детальне вікно проекту представлено на рисунку 2.5.

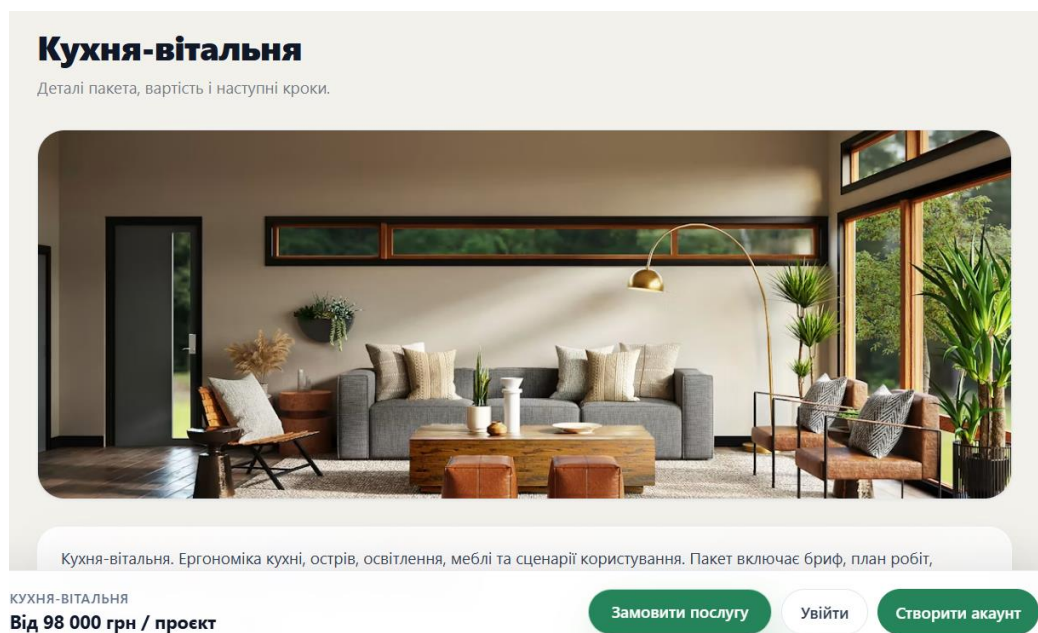


Рисунок 2.5 – Детальне вікно проекту

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

Панель «Відгуки», для перегляду відгуків клієнтів представлено на рисунку

2.6

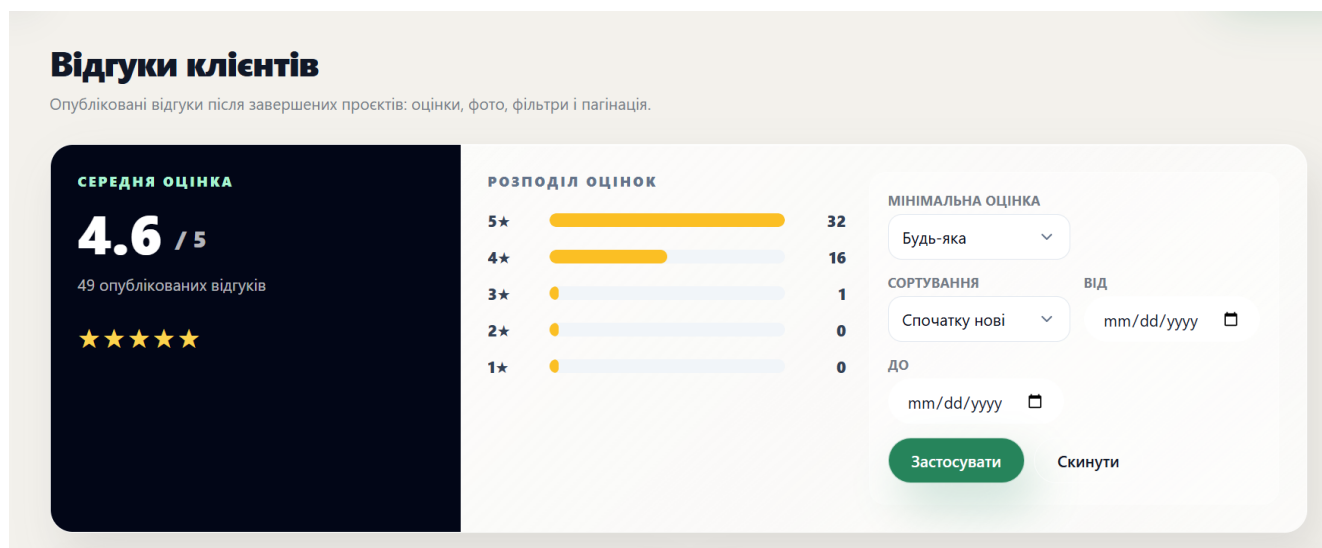


Рисунок 2.6 – Панель «Відгуки»

Панель «Команда», для перегляду усіх працівників представлено на рисунку

2.7.

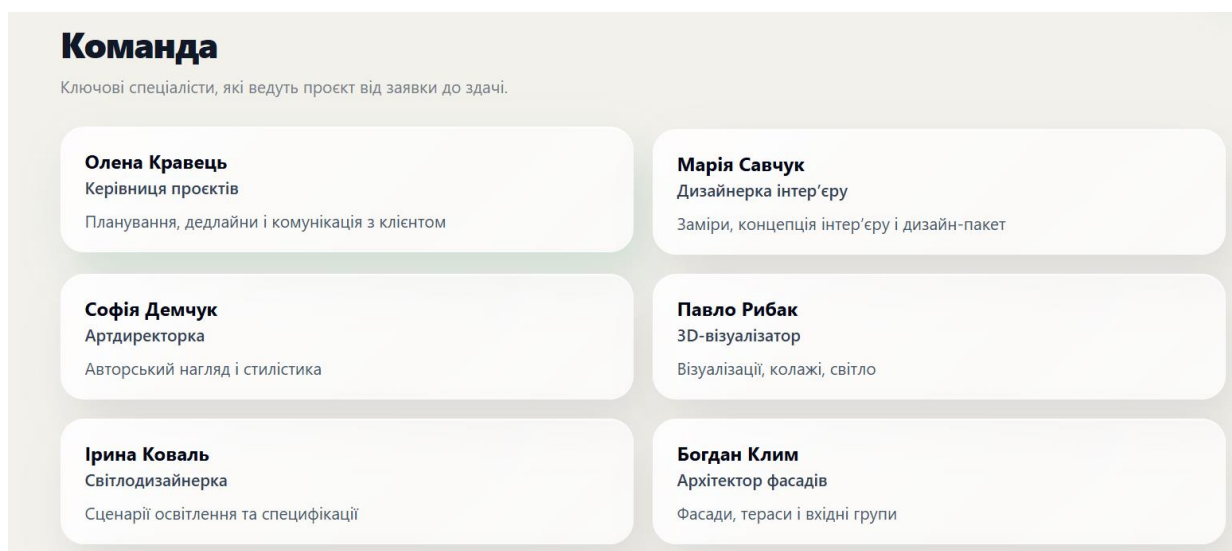


Рисунок 2.7 – Панель «Команда»

Панель «Контакти» представлено на рисунку 2.8.

Контакти

Опишіть простір, задачу і бажані терміни. Ми відповідаємо протягом робочого дня.

Лінія студії: +380 67 000 00 00 · hello@tailored.design
Графік: понеділок-п'ятниця, 09:00-18:00

Ім'я та прізвище

Email

Телефон (необов'язково)

Деталі проєкту

Локація, обсяг робіт, бажані терміни, бюджет...

Рисунок 2.8 – Панель «Контакти»

Вхід в ролі звичайного користувача представлено на рисунку 2.9.

Tailored
DESIGN SOLUTIONS

Оформити заявку

Увійдіть у клієнтський кабінет або створіть новий акаунт.

EMAIL
client@tailored.demo

ПАРОЛЬ
.....

Увійти →

Немає акаунта? Створити профіль клієнта

Рисунок 2.9 – Авторизація

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

Головну панель користувача представлено на рисунку 2.10.

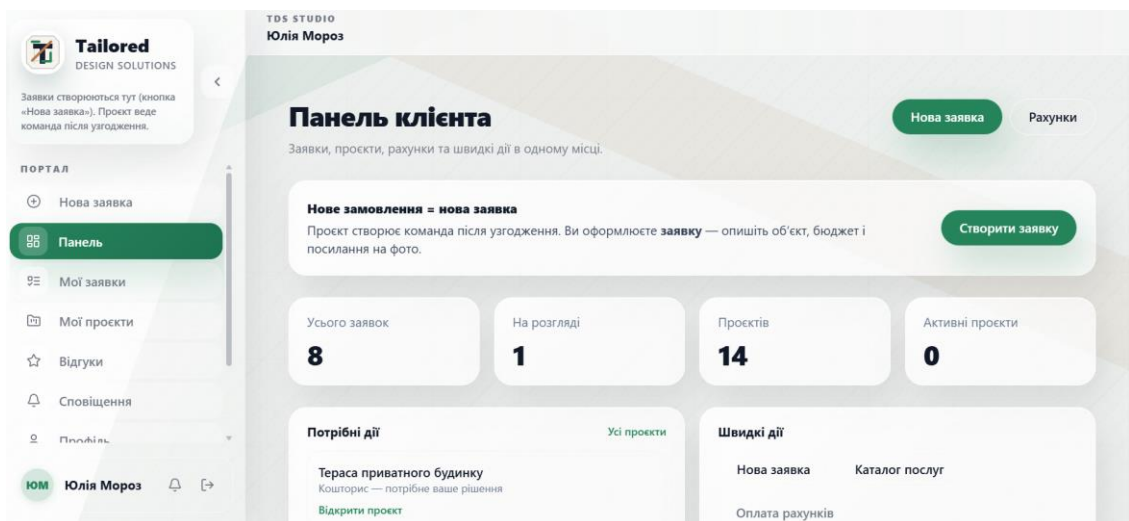


Рисунок 2.10 – Головна панель

Створення нової заявки представлено на рисунку 2.11.

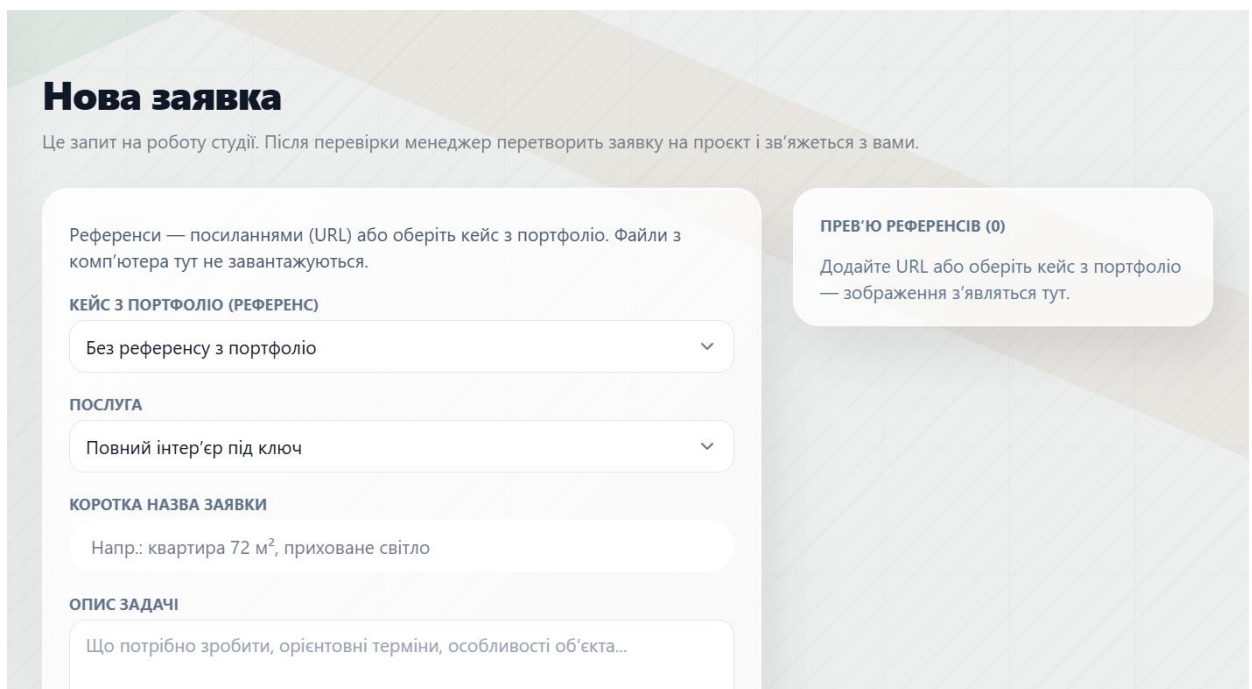


Рисунок 2.11 – Нова заявка

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Отриманий результат представлено на рисунку 2.12.

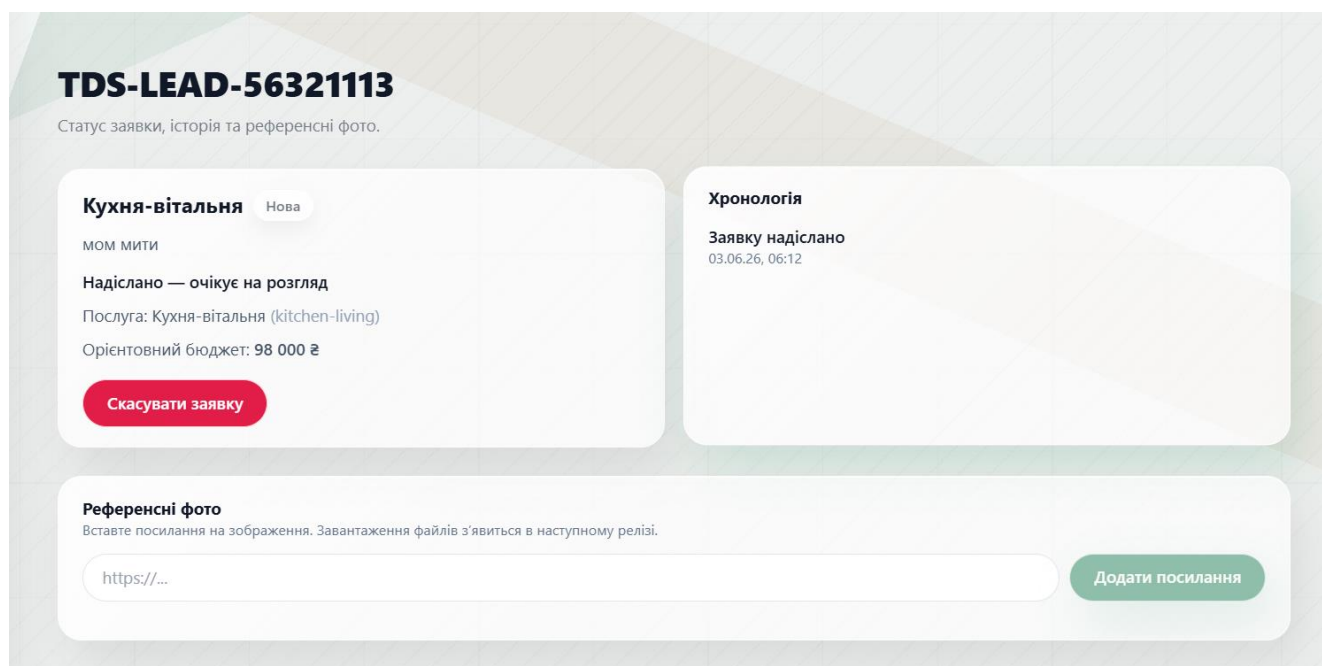


Рисунок 2.12 – Вікно управління заявкою

Перегляд усіх власні проєкти представлено на рисунку 2.13.

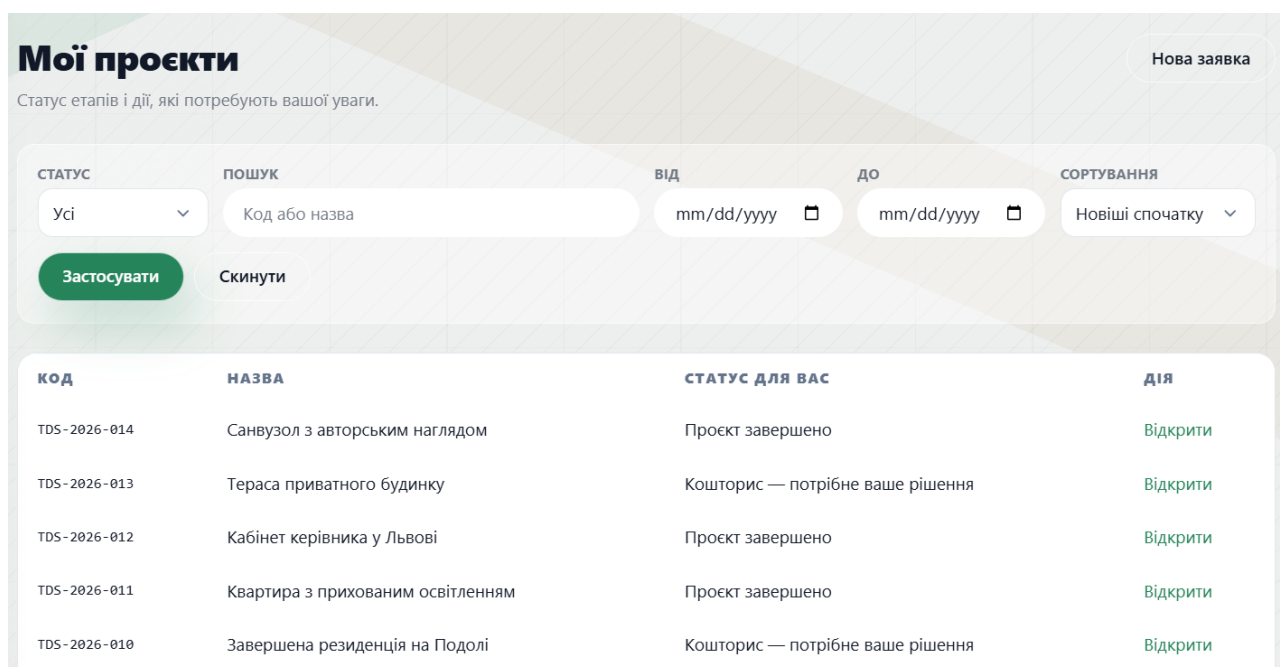


Рисунок 2.13 – вкладка «Мої проєкти»

Вхід на вкладку «Сповіщення» представлено на рисунку 2.14.

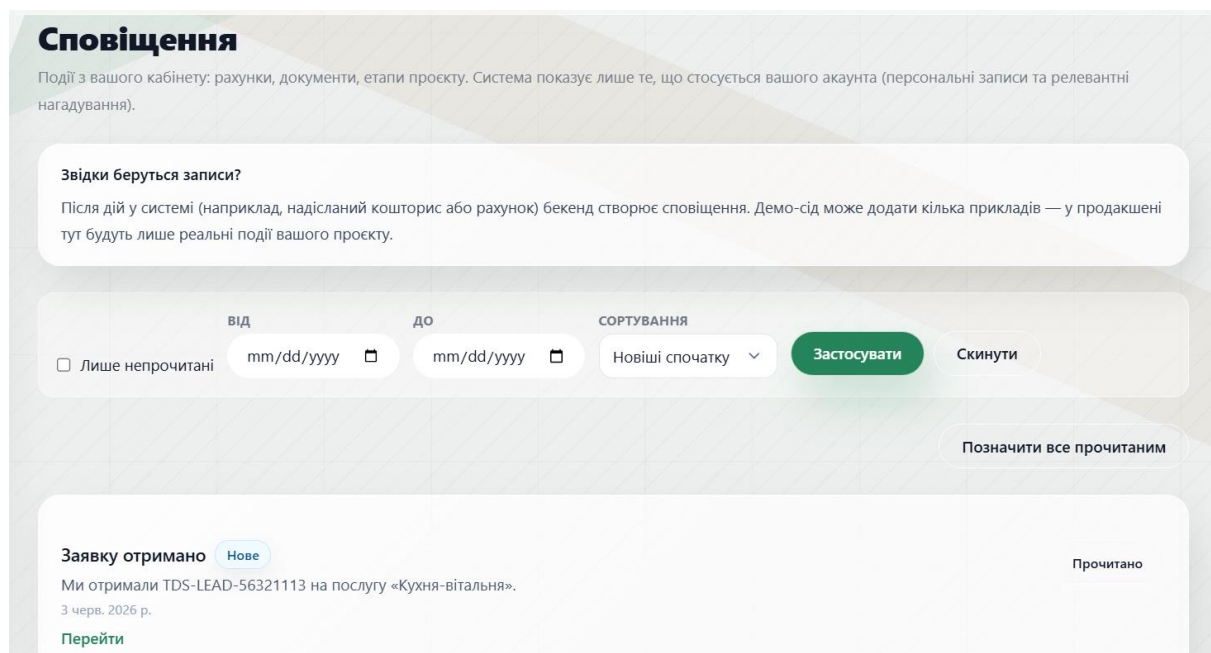


Рисунок 2.14 – вкладка «Сповіщення»

Вхід на вкладку «Профіль» та зміни його представлено на рисунку 2.15.

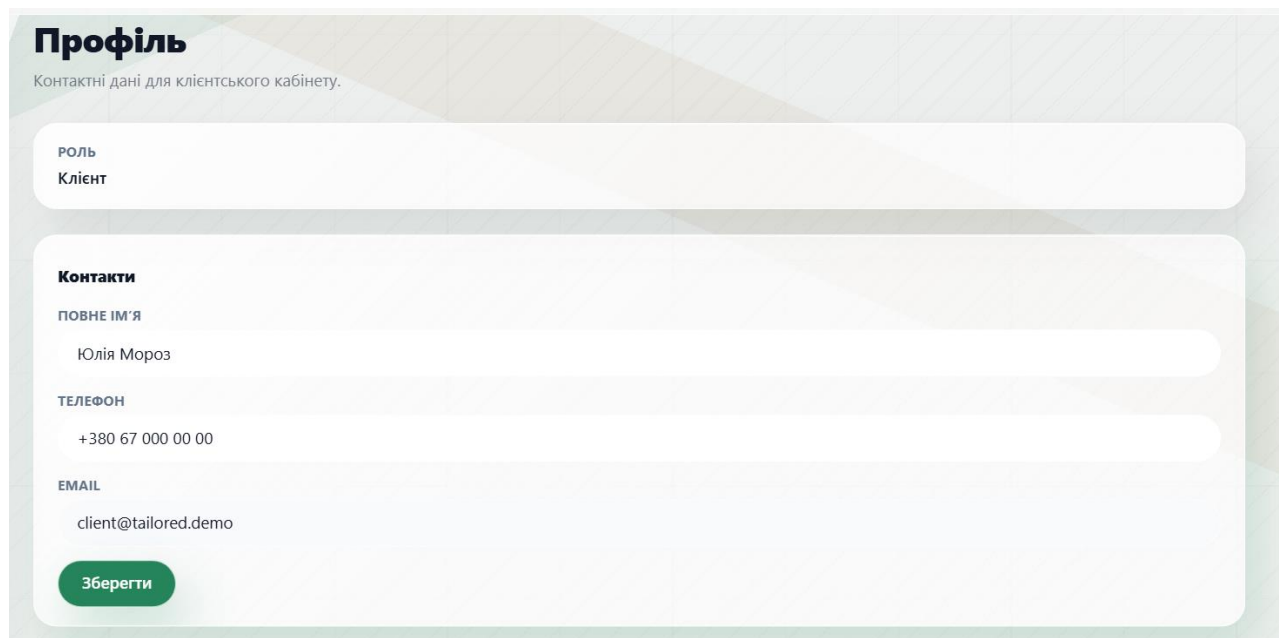


Рисунок 2.15 – вікно «Профіль»

Вхід на вкладку «Рахунки» та перегляд її представлено на рисунку 2.16.

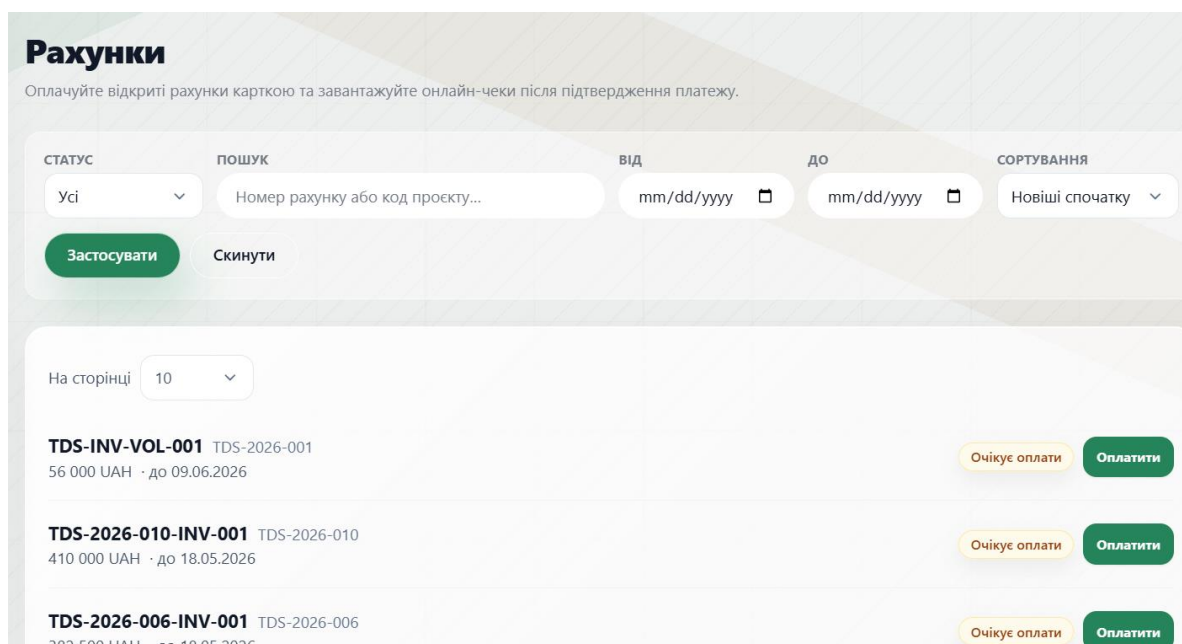


Рисунок 2.16 – Вкладка «Рахунки»

Вхід на вкладку «Онлайн-чеки» та перегляд її представлено на рисунку 2.17.

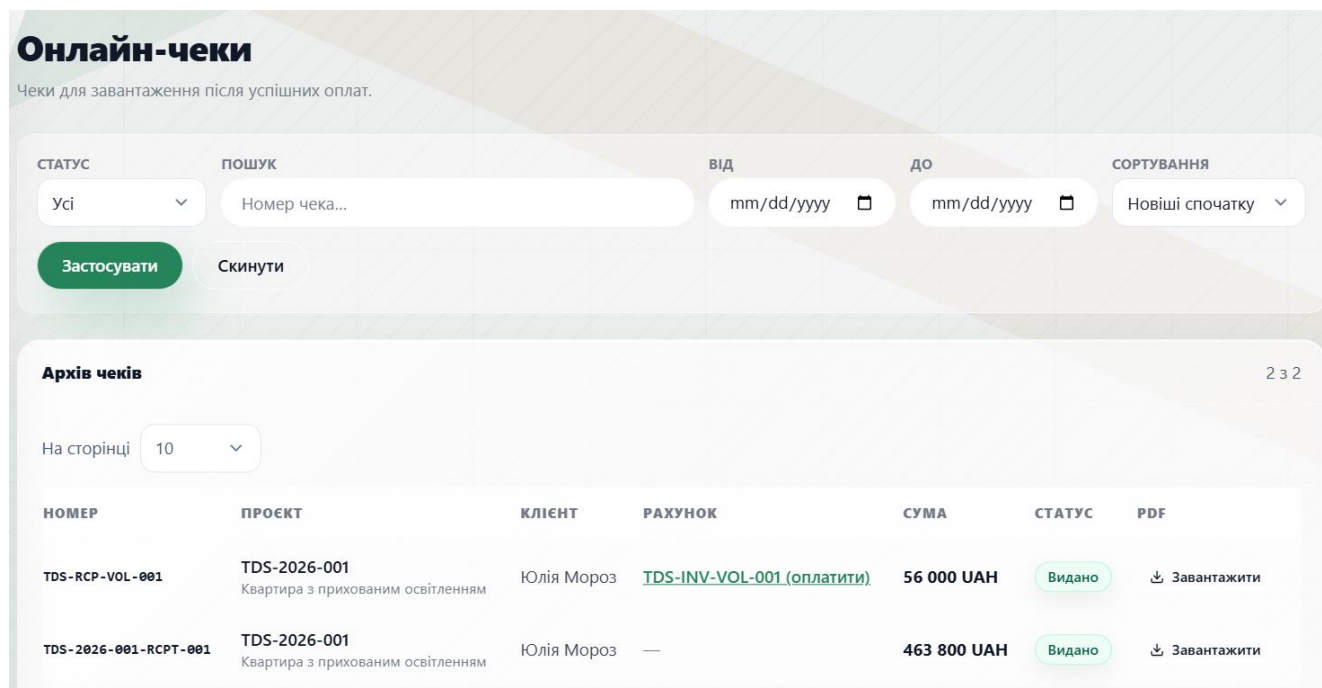


Рисунок 2.17 – Вкладка «Онлайн-чеки»

Вихід із акаунта звичайного користувача та зміна ролі на дизайнера представлено на рисунку 2.18.

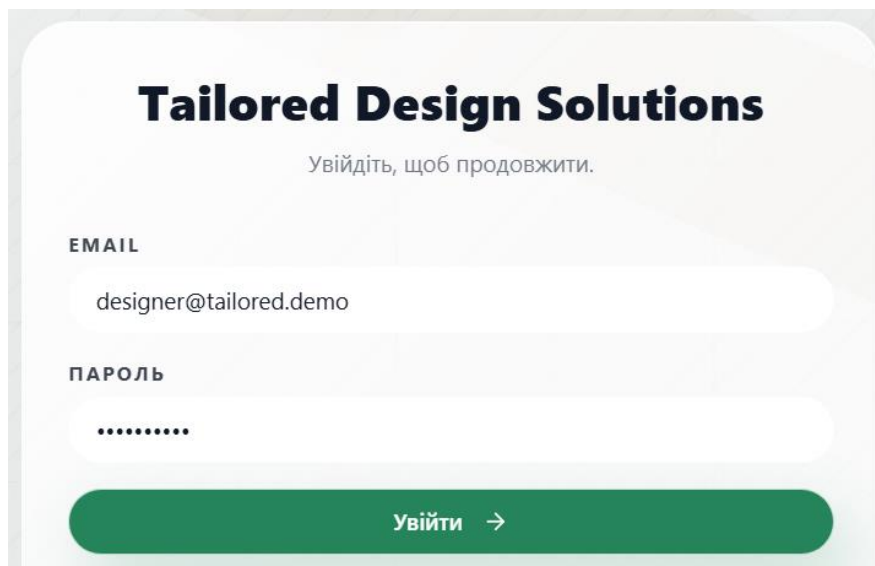


Рисунок 2.18 – Авторизація

Вхід на вкладку «Моя робота» представлено на рисунку 2.19.

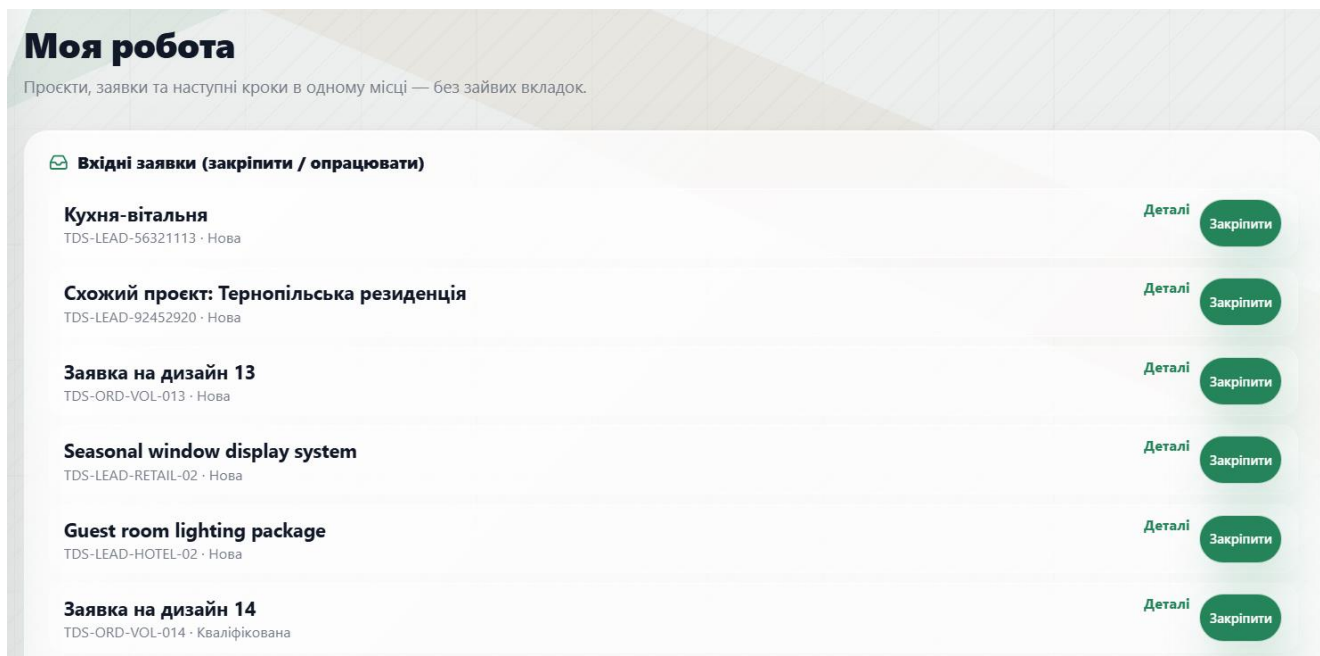


Рисунок 2.19 – Вікно «Моя робота»

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

Вхід на вкладку «Заявки» представлено на рисунку 2.20.

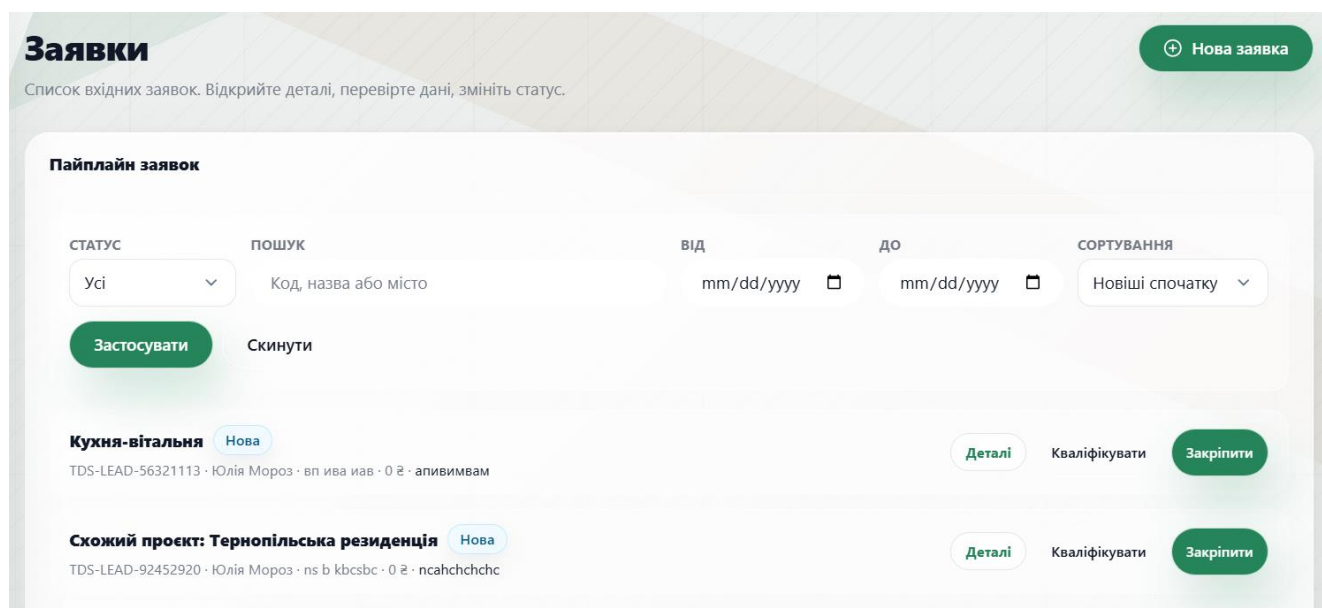


Рисунок 2.20 – Вікно «Заявки»

Вхід на вкладку «Проєкти» представлено на рисунку 2.21.

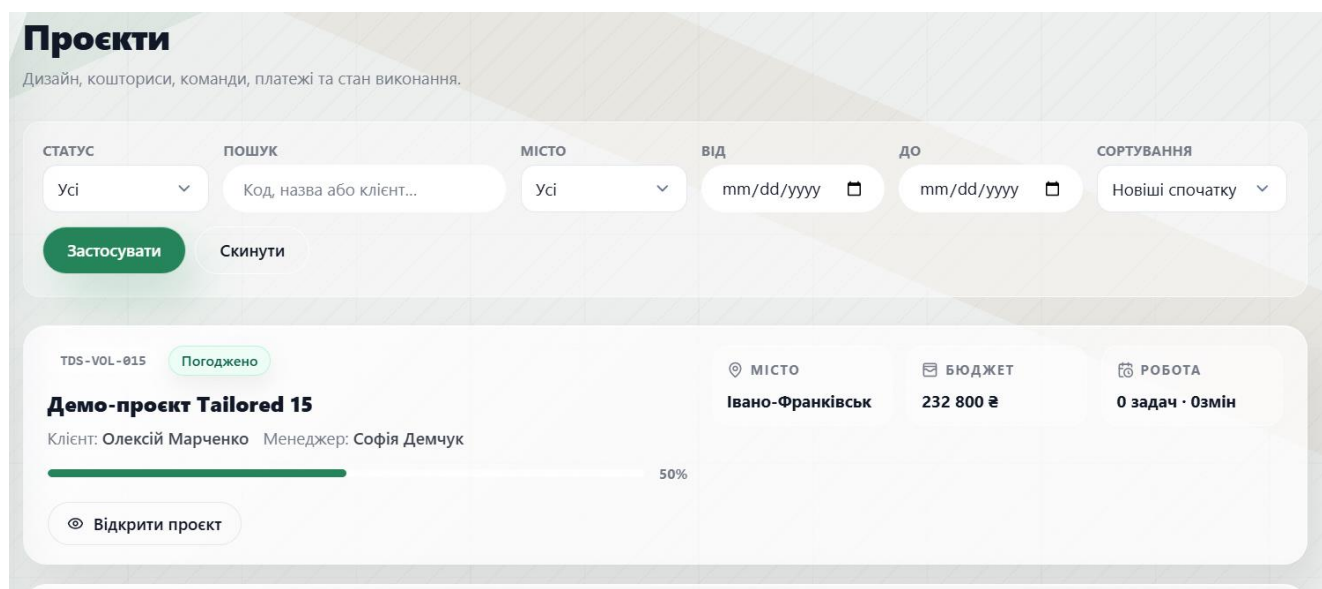


Рисунок 2.21 – Вікно «Проєкти»

Вхід на вкладку «Каталог послуг» представлено на рисунку 2.22.

Каталог послуг

Керуйте публічними послугами. Дизайнери бачать свої записи, адміністратори бачать усе.

Записи

Нова послуга

Оберіть або створіть

SLUG

НАЗВА

КОРОТКИЙ ОПИС

БАЗОВА ЦІНА

ОДИНИЦЯ ЦІНИ

0

project

СТИЛІ (ЧЕРЕЗ КОМУ)

ПОРЯДОК СОРТУВАННЯ

АКТИВНА

Рисунок 2.22 – Вікно «Каталог послуг»

Вхід на вкладку «Заміри» представлено на рисунку 2.23.

Заміри

Додати замір

Заміри по зонах для розрахунку кошторисів. Фільтри та пагінація працюють через API (проект, текстовий пошук по зоні/нотатках).

Фільтри та пошук

ПОШУК

ПРОЕКТ

НА СТОРІНЦІ

ВІД

ДО

Напр. вітальня, ніша...

Усі проекти

25

mm/dd/yyyy

mm/dd/yyyy

СОРТУВАННЯ

Новіші спочатку

Застосувати

Скинути

ЗОН ЗА ФІЛЬТРОМ (УСЬОГО)

ПІДЛОГА ЗА ФІЛЬТРОМ (М²)

СТІНИ ЗА ФІЛЬТРОМ (М²)

26

552.6 м²

1037.3 м²

TDS-2026-006 5 зон на сторінці Exterior terrace and entrance upgrade

ЗОНА ПІДЛОГА М² СТІНИ М² ВИСОТА М НОТАТКИ

Рисунок 2.23 – Вкладка «Заміри»

Вхід в ролі звичайного користувача представлено на рисунку 2.24.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

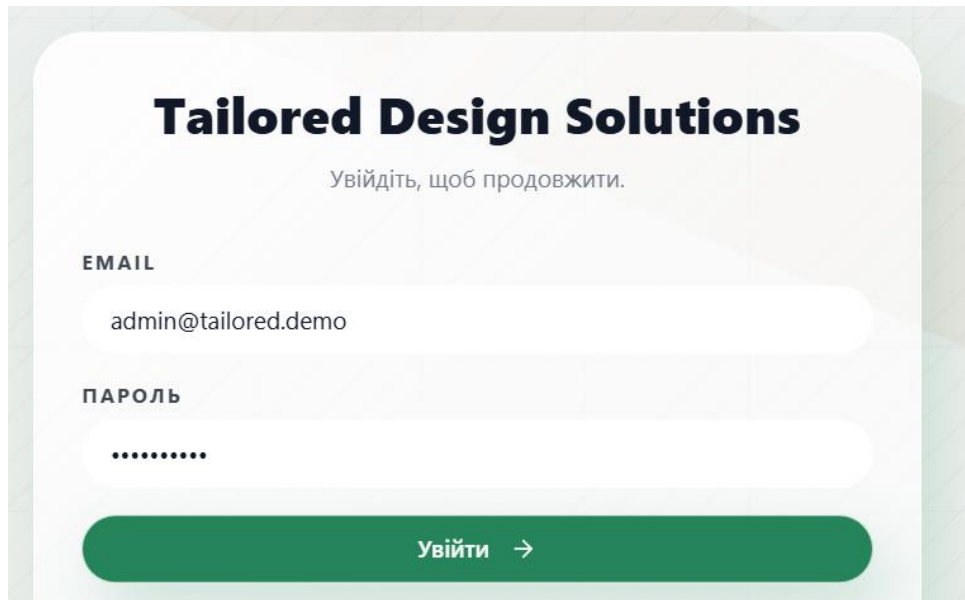


Рисунок 2.24 – Авторизація

Вхід на вкладку «Огляд» представлено на рисунку 2.25.

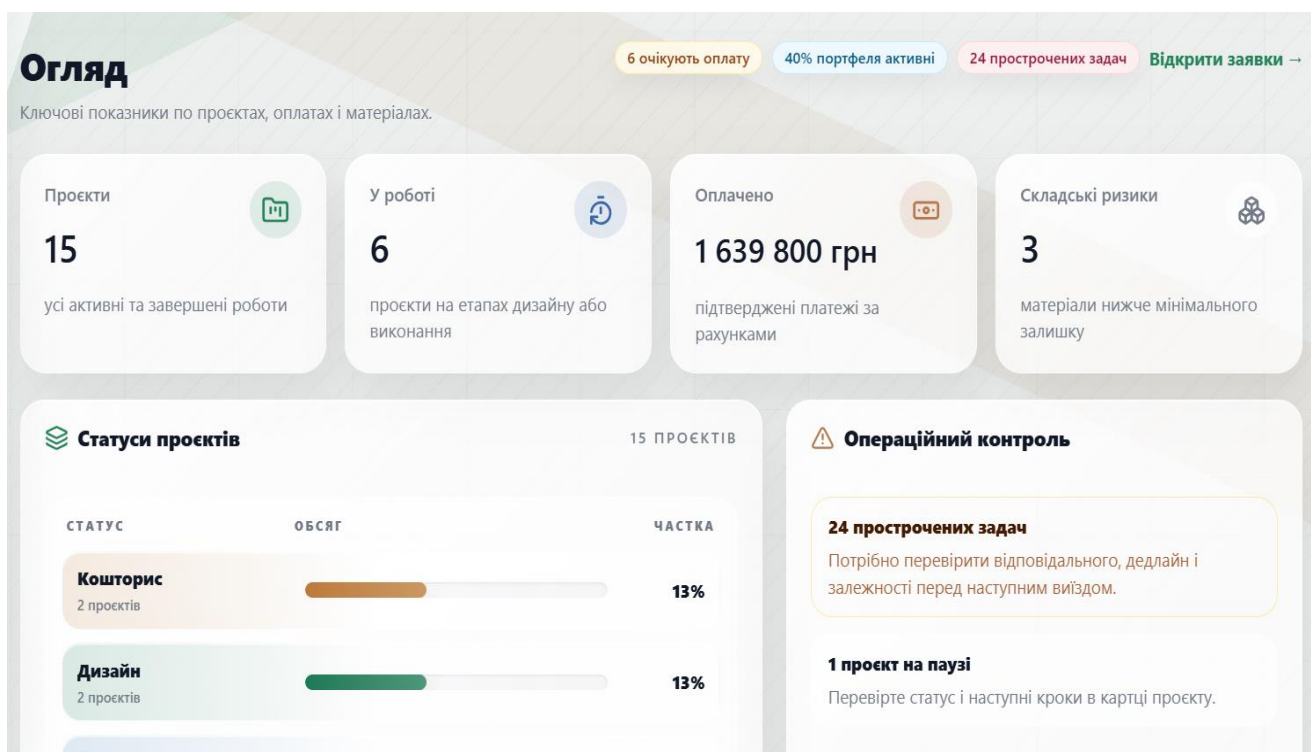


Рисунок 2.25 – Вкладка «Огляд»

Вхід на вкладку «Аналітика» представлено на рисунку 2.26

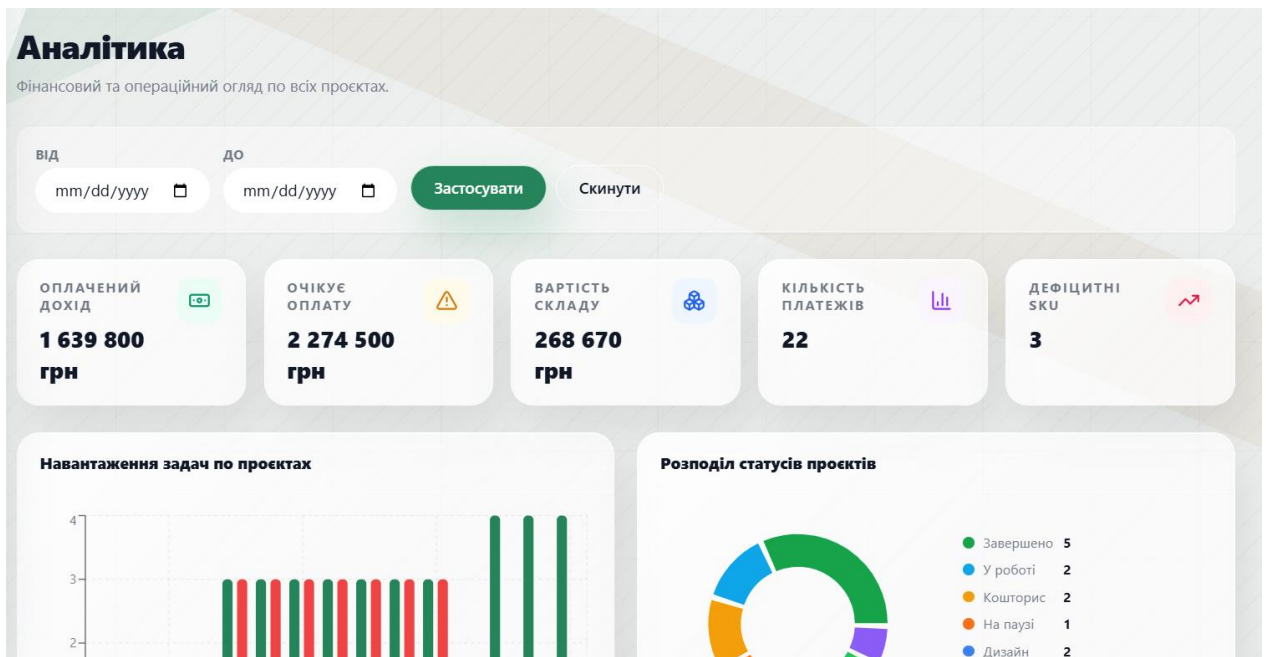


Рисунок 2.26 – Вкладка «Аналітика»

Вхід на вкладку «Звіти» представлено на рисунку 2.27

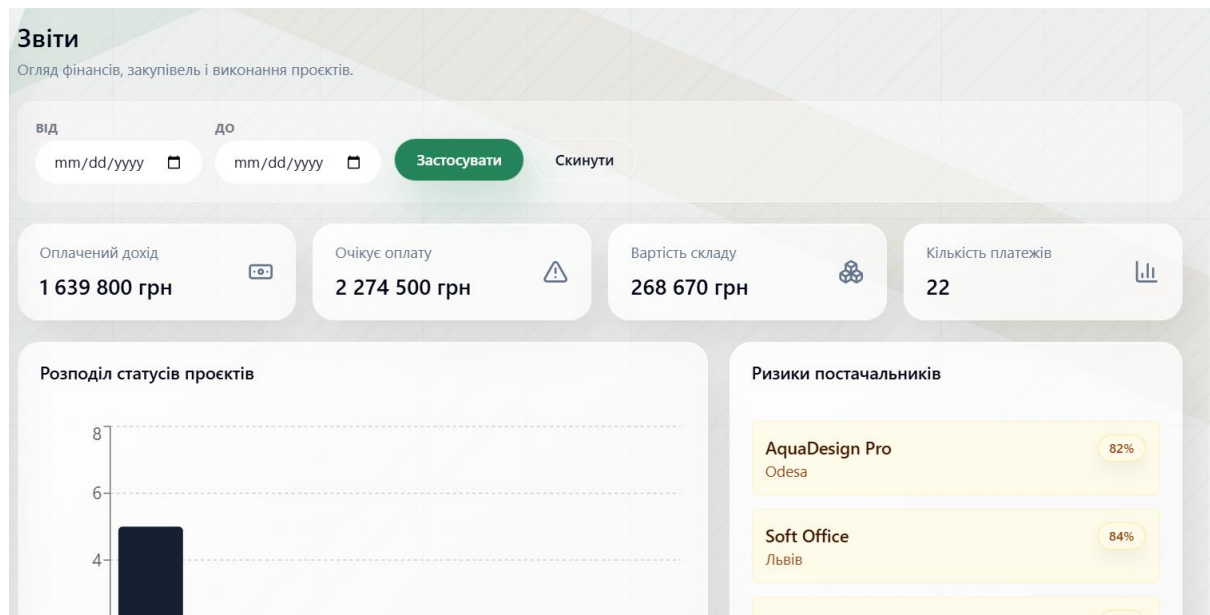


Рисунок 2.27 – Вікно «Звіти»

Вхід на вкладку «Матеріали та постачальники» представлено на рисунку 2.28

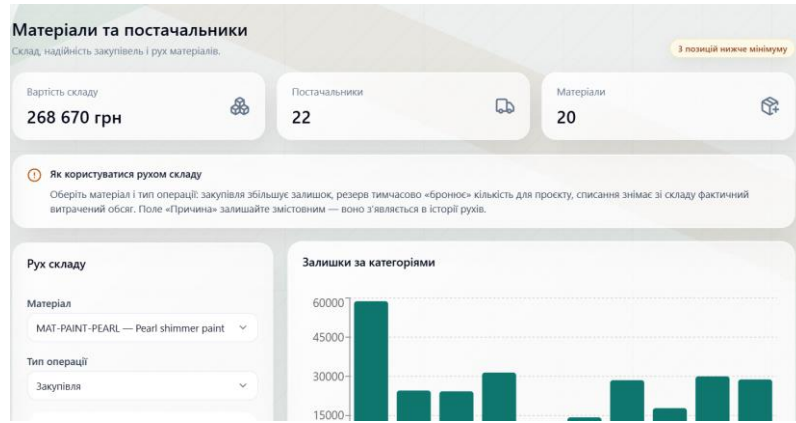


Рисунок 2.28 - Вкладка «Матеріали та постачальники»

Вхід на вкладку «Модерація відгуків» представлено на рисунку 2.29

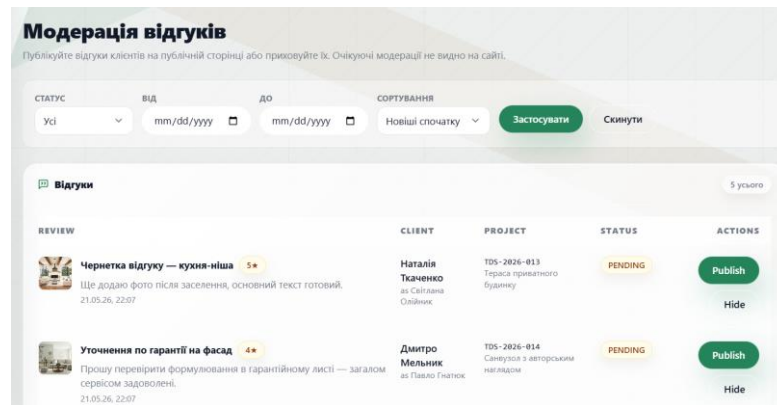


Рисунок 2.29 - Вкладка «Модерація відгуків»

Вхід на вкладку «Користувача та ролі» представлено на рисунку 2.30

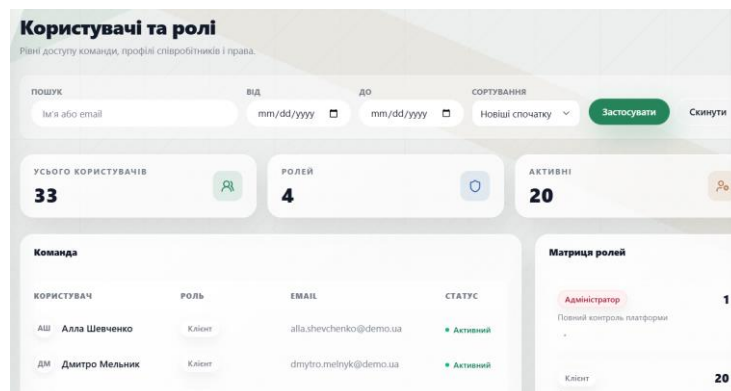


Рисунок 2.30 - Вкладка «Користувача та ролі»

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розгортання вебзастосунку

Розгортання вебсервісу «Interiorix» передбачає не лише локальне використання, але й надання доступу до сервісу з мережі Інтернет. Для цього потрібно забезпечити наявність білої (публічної) IP-адреси, яка дозволить користувачам підключатись до сервера з будь-якої точки світу. У випадку розгортання на віртуальному приватному сервері (VPS) або хмарному хостингу, така адреса зазвичай надається автоматично. Важливо також відкрити відповідні порти у фаєрволі чи маршрутизаторі: 3000 – для HTTP/HTTPS-доступу до інтерфейсу користувача (frontend), 4000 – для API-запитів до бекенду (якщо вони не проксіювані), порт `mongodb://mongodb:27017/tailored` бази даних MongoDB відкривати не рекомендується з міркувань безпеки, замість цього бажано використовувати VPN або тунелювання.

Після складання клієнтської частини вебсервісу за допомогою команди `pnpm build`, необхідно налаштувати вебсервер, який оброблятиме запити користувачів і віддаватиме статичні ресурси. Найбільш доцільним вибором для цього є Nginx – високопродуктивний вебсервер, який легко інтегрується з Docker та забезпечує проксійний доступ до бекенду. У типовій конфігурації вебсервер слухає порт 80 та обробляє запити до кореневого шляху `/`, повертаючи файл `index.html`, а запити до `/api/` перенаправляє на бекенд, що працює на порту 4000. Це дозволяє приховати деталі реалізації серверної частини та централізовано обробляти трафік.

Щоб вебсервіс був доступним за доменним іменем, необхідно зареєструвати домен та додати до нього DNS-запис типу A, що вказуватиме на публічну IP-адресу сервера. Для забезпечення захищеного з'єднання обов'язково потрібно налаштувати SSL-сертифікат. Найпростіший і безкоштовний спосіб зробити це – скористатись сервісом Let's Encrypt. За допомогою інструмента Certbot можна автоматично отримати та оновлювати сертифікати, які інтегруються з Nginx.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

3.2 Інструкція з наповнення вебзастосунку

Ефективне функціонування вебсервісу «Interiorix» безпосередньо залежить від правильного і систематичного наповнення системи актуальними, структурованими та достовірними даними. Інструкція з наповнення є ключовим компонентом у впровадженні програмного продукту, оскільки вона визначає алгоритм додавання нових проєктів, наповнення інформаційних блоків, управління користувачьким контентом, фотоматеріалами, а також формування супровідної інформації, необхідної для коректного функціонування платформи.

Першочерговим кроком є створення облікового запису адміністратора, який буде відповідальним за внесення об'єктів до бази даних. Після авторизації через захищену панель управління, адміністратор отримує доступ до функціоналу додавання нового дизайнерського рішення. На цьому етапі потрібно натиснути кнопку «Додати проєкт», що відкриває форму з кількома секціями для заповнення.

У першу чергу заповнюється загальна інформація про об'єкт. До цієї категорії входять такі обов'язкові поля: тип дизайну (житловий чи комерційний), кількість кімнат, загальна площа (м²), вартість,. Варто звернути увагу на правильність написання всіх полів, адже саме за ними користувачі будуть шукати потрібні рішення.

Далі слідує секція опису об'єкта, де необхідно надати детальну текстову інформацію про деталі дизайну, тощо. Важливо подавати цю інформацію у зрозумілій, грамотній формі, без помилок, із акцентом на ключових перевагах конкретної пропозиції. Це значно підвищує зацікавленість користувача, рівень довіри до платформи та шанси на замовлення.

Наступним етапом є завантаження фотофайлів. Система дозволяє прикріпити до 20 фотографій у форматах .jpg, .jpeg або .png. Зображення мають бути чіткими, добре освітленими, бажано зробленими у денний час. Для кожного об'єкта можна вказати «головне фото», яке буде відображатися на прев'ю картки у загальному списку.

Після завершення заповнення форми адміністратор має можливість

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

переглянути попередній вигляд оголошення, відредагувати або одразу опублікувати його. Система перевіряє заповненість обов'язкових полів і не дозволить зберегти оголошення без них. Опублікований об'єкт зберігається у базі даних та стає доступним на головній сторінці сайту, у відповідних категоріях та результатах пошуку.

Окрему увагу варто приділити регулярному оновленню та верифікації даних. Застарілі або вже неактуальні оголошення повинні видалятися або позначатися як «неактивні». Це забезпечує якість і правдивість інформації на платформі, що є однією з головних переваг «Interiorix». Адміністратори можуть скористатися внутрішнім модулем звітності для перегляду стану активності кожного оголошення та отримання аналітики щодо переглядів, заявок, відгуків.

Важливою складовою наповненням вебсервісу є управління категоріями дизайнерських проєктів. Адміністратор має можливість створювати нові категорії, редагувати їх описи та налаштовувати відображення на головній сторінці. Це дозволяє структурувати контент за напрямками дизайну, такими як сучасний стиль, мінімалізм, класичний стиль, лофт, скандинавський стиль та інші. Завдяки цьому користувачі можуть швидше знаходити дизайнерські рішення, які відповідають їхнім потребами та вподобаннями.

Окремий функціональний модуль призначений для роботи з відгуками та заявками клієнтів. Після перегляду проєкту користувачі можуть залишати власні відгуки та оцінки, що сприяють формуванню репутації дизайнерського агентства. Адміністратор має можливість модерувати коментарі, видаляти некоректний контент та аналізувати рівень зацікавленості відвідувачів окремими дизайнерськими рішеннями. Це дозволяє підвищувати якість контенту та забезпечувати комфортну взаємодію між клієнтами й компанією.

Дотримання вищенаведених інструкцій дозволить забезпечити якісне представлення послуг, формувати довіру серед клієнтів та підтримувати репутацію сучасного, технологічного і відповідального дизайнерського агентства «Interiorix», на ринку створення дизайнерських рішень для інтер'єру та екстер'єру.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

3.3 Інструкція з популяризації та підтримки вебзастосунку

Після завершення розробки та впровадження вебзастосунку «Interiorix» важливою частиною життєвого циклу продукту стає його популяризація та подальша технічна підтримка. Без належної уваги до просування і супроводу навіть найкраще технічне рішення може залишитися поза увагою цільової аудиторії, що призведе до низького рівня використання платформи. Саме тому необхідно комплексно підійти до завдань маркетингового просування, створення впізнаваності бренду, підтримки користувачів, а також стабільної роботи системи.

Популяризація вебсервісу передбачає реалізацію стратегії цифрового маркетингу, яка включає рекламні кампанії в соціальних мережах (Facebook, Instagram, TikTok), SEO-оптимізацію сайту, контекстну рекламу в пошукових системах (Google Ads), співпрацю з блогерами та тематичними порталами про дизайн, а також розсилку інформаційних матеріалів для потенційних партнерів. Розробка унікального бренду (логотип, кольорова гама, стиль візуалізації), а також створення якісного контенту (статті, відеоогляди, інтерв'ю з дизайнерами, аналітика ринку) сприяють формуванню довіри до сервісу серед нових користувачів.

Крім того, важливо регулярно публікувати звіти про нові функції та оновлення платформи, демонструючи постійний розвиток і адаптацію до потреб ринку.

Ще одним напрямом популяризації є локальні рекламні кампанії у великих містах України – зовнішня реклама на біл-бордах та на зданих об'єктах, участь у виставках і конференціях з дизайну, співпраця з будівельними компаніями та дизайнерськими асоціаціями. Також варто створити канал на YouTube з оглядами готових об'єктів, інтерв'ю з постачальниками, дизайнерськими порадами порадами тощо. Важливою перевагою «Interiorix» є акцент на якість робіт та прозору інформацію, що варто активно просувати як конкурентну перевагу перед іншими платформами.

Окрім заходів з просування, велике значення має налагодження якісної

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

системи підтримки користувачів. На сайті повинна бути доступна форма зворотного зв'язку, розділ із часто задаваними питаннями (FAQ), онлайн-чат з підтримкою, а також система повідомлень для оперативного інформування про зміну статусу заявки. Зручність і швидкість отримання допомоги значно підвищує лояльність користувачів.

Також необхідно забезпечити постійний технічний моніторинг роботи платформи. Це включає щоденний контроль доступності сайту, перевірку коректності API-запитів, регулярне оновлення безпеки, створення резервних копій бази даних, тестування продуктивності при зростанні навантаження. Для цього варто впровадити систему автоматичного логування помилок та аналітики дій користувачів з метою виявлення «вузьких місць» у логіці застосунку. Для моніторингу стану серверів рекомендується використовувати такі інструменти, як Grafana, Prometheus.

З метою довгострокової підтримки системи та розвитку її функціоналу, доцільно сформувавши дорожню карту розвитку, яка міститиме перелік запланованих функцій, орієнтовні терміни реалізації та пріоритетність. Це дозволить не тільки керувати очікуваннями користувачів, а й забезпечити стратегічне планування технічних робіт. Зворотний зв'язок із користувачами — стане основою для формування пріоритетів розвитку.

Отже, ефективне функціонування та популярність вебсервісу «Interiorix» напряму залежить не лише від якості реалізації програмного забезпечення, але й від систематичної роботи з просуванням, підтримки та вдосконалення продукту. Завдяки комплексному підходу, який включає маркетингові активності, технічний супровід і постійне вдосконалення функціоналу, вебсервіс може зайняти провідну позицію на українському ринку онлайн-сервісів із створення власних дизайнерських рішень для інтер'єру та екстер'єру.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

4. ЕКОНОМІЧНИЙ РОЗДІЛ

«Interiorix» прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є вебсервіс «Interiorix».

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 та термін окупності.

4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР

У цьому підрозділі розглянемо основні етапи технологічного процесу для розробки вебсервісу «Interiorix». Розробка такого продукту являє собою послідовний процес, що складається з низки взаємопов'язаних операцій, кожна з яких потребує певних витрат робочого часу. Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю. Такий підхід дає наочне уявлення про структуру затрат часу, дозволяє раціонально спланувати весь цикл створення продукту та обчислити загальну тривалість проведення науково-дослідної роботи. Чітке визначення переліку робіт та їхньої тривалості також допомагає оцінити завантаженість виконавців і завчасно виявити операції, що потребують найбільших затрат ресурсів.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

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.
	Планування та аналіз	Кер. проєкту Рm	
		Інженер (11)	8
	Розробка технічного завдання	Кер. проєкту (Рm)	7
		Інженер (11)	
	Дизайн інтерфейсу	Інженер (11)	
		Інженер (12)	
	Розробка функціоналу	Інженер (11)	4
	Тестування та відладка	Тестувальник	8
	Документування	Інженер (11)	
	Розгортання та підтримка	Інженер (12)	
	Управління проєктом	Кер. проєкту (Рm)	
Разом			

Сумарний час виконання операцій технологічного процесу становить 166 години.

4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки вебсервісу «Inreriorix».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та діяльності підприємства.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

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 (Pm) – 450 грн./год, інженера (I2) – 272 грн./год.), інженера (I1) – 113 грн./год., тестувальник – 100 грн./год.; K_r – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

$$\text{Керівника проекту (Pm)} \quad Z_{\text{осн}2} = 40 * 450 = 18\,000 \text{ грн.}$$

$$\text{Інженера (I2)} \quad Z_{\text{осн}4} = 40 * 272 = 10\,880 \text{ грн.}$$

$$\text{Інженера (I1)} \quad Z_{\text{осн}3} = 78 * 113 = 8\,814 \text{ грн.}$$

$$\text{Тестувальник} \quad Z_{\text{осн}4} = 8 * 100 = 800 \text{ грн.}$$

Сумарна основна заробітна плата становить

$$Z_{\text{осн}} = 18\,000 + 10\,880 + 8\,814 + 800 = 38\,494 \text{ грн.}$$

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:

$$\text{Керівника проекту} \quad Z_{\text{дод}2} = 18\,000 * 0,1 = 1\,800 \text{ грн.}$$

$$\text{Інженера (I2)} \quad Z_{\text{дод}3} = 10\,880 * 0,1 = 1\,088 \text{ грн.}$$

$$\text{Інженера (I1)} \quad Z_{\text{дод}4} = 8\,814 * 0,1 = 881 \text{ грн.}$$

$$\text{Тестувальник} \quad Z_{\text{дод}4} = 800 * 0,1 = 80 \text{ грн.}$$

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод}} = 1\,800 + 1\,088 + 881 + 80 = 3\,849 \text{ грн.}$$

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

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

$$V_{\text{о.п.}} = 38\,494 + 3\,849 = 42\,343 \text{ грн.}$$

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

$$V_{\text{есв}} = V_{\text{оп}} * 0,22 \quad (4.4)$$

$$V_{\text{есв}} = 42\,343 * 0,22 = 9\,315 \text{ грн.}$$

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

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.			Дода ткова зароб ітна плата , грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. 6 = 3+4+5
		Тарифн а ставка, грн.	К-сть годин	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Кер. проєкту (Pm)	450	40	18 000	1 800	4 359	24 159
2	Інженера (I2)	272	40	10 880	1088	2 633	14 601
3	Інженера (I1)	113	78	8 814	881	2 133	11 827
4	Тестувальник	100	8	800	80	194	1 073
Разом				38 494	3 849	9 315	51 658

Отже, загальні витрати на оплату праці становлять 51 658 грн.

4.3. Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_{\text{в}} = W * T * S \quad (4.5)$$

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.

$$Z_{\text{ек}} = 0,82 * 166 * 15,94 = 2 169 \text{ грн.}$$

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

Витрати на електроенергію становлять 2 169 грн.

4.4. Розрахунок суми амортизаційних відрахувань розробки вебсервісу «Inreriorix»

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення

Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{БВ * Н_A}{100\%} * T \quad (4.6)$$

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

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

Н_А – норма амортизації, 0,04 %.

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

$$A = \frac{50\ 000,00 * 0,04}{150} * 166 = 2\ 213 \text{ грн.}$$

4.5. Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

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

де: H_B – накладні витрати.

$$H_B = 42\,343 * 0,4 = 16\,937 \text{ грн.}$$

4.6. Складання кошторису витрат та визначення собівартості розробки вебсервісу «Inreriorix»

Для складання кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.4.

Таблиця 4.4 - Кошторис витрат розробки вебсервісу «Inreriorix»

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	51 658	71
2.	Витрати на електроенергію	2 169	3
3.	Амортизаційні відрахування	2 213	3
4.	Накладні витрати	16 937	23
5.	Собівартість	72 977	100

Собівартість (C_B) НДР розраховуємо за формулою:

$$C_B = V_{o.p.} + V_{c.z} + 3e + A + H_B \quad (4.8)$$

Отже, собівартість дорівнює $C_B = 72\,977$ грн.

4.7. Розрахунок ціни розробки вебсервісу «Inreriorix»

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:

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

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

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

$$Ц = 72\,977 * (1 + 0,3) * (1 + 0,2) = 113\,844 \text{ грн.}$$

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.

Розрахунок прибутку визначається з а формулою:

$$П = Ц - C_B \quad (4.10)$$

де: $Ц$ – ціна розробки, грн.; C_B – собівартість розробки, грн.

$$П = 113\,844 - 72\,977 = 40\,867 \text{ грн.}$$

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності ($T_{ок}$).

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

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

$$ЧТВ = -72\,977 + \frac{40\,867}{(1+0,1)^1} + \frac{40\,867}{(1+0,1)^2} + \frac{40\,867}{(1+0,1)^3} = 28\,654 \text{ грн}$$

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

Термін окупності визначається за формулою:

$$T_{ок} = T_{пв} + \frac{H_B}{\Gamma_{пр}} \quad (4.12)$$

де: $T_{пв}$ – період до повного відшкодування витрат, років; H_B – невідшкодовані витрати на початок року, грн.; $\Gamma_{пр}$ – грошовий потік на початок року, грн.

$$T_{ок} = 2 + \frac{2\,051}{40\,867} = 2,1 \text{ р.}$$

Всі дані внесемо в зведену таблицю 4.5.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.5 – Техніко-економічні показники розробки вебсервісу «Inrerioгіx»

№ п/п	Показник	Значення
1.	Собівартість, грн.	72 977
2.	Плановий прибуток або грошовий потік, грн.	40 867
3.	Ціна, грн.	113 844
4.	Чиста теперішня вартість, грн.	28 654
5.	Термін окупності, рік	2,1

Прибутковість проекту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом 2,1 року. Отже, на основі отриманих показників можна зробити висновок, що розробка вебсервісу «Inrerioгіx» є доцільною з економічної точки зору.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Поняття праці та охорони праці

Праця є однією з найважливіших форм діяльності людини, спрямованою на створення матеріальних і духовних цінностей, забезпечення власних потреб та розвитку суспільства. Завдяки праці людина отримує можливість забезпечувати належний рівень життя, реалізовувати свої здібності та професійні навички. Праця відіграє важливу роль у розвитку економіки держави, оскільки саме трудова діяльність є основою функціонування підприємств, установ і організацій. У сучасному суспільстві праця є не лише джерелом доходу, а й важливим фактором соціального розвитку особистості. Конституція України гарантує кожному громадянину право на працю, що включає можливість заробляти собі на життя працею, яку людина вільно обирає або на яку погоджується. Водночас право на працю нерозривно пов'язане з правом на безпечні та здорові умови праці. Саме тому держава приділяє значну увагу питанням охорони праці та створенню безпечного виробничого середовища. Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності працівників у процесі трудової діяльності. Основною метою охорони праці є запобігання виробничому травматизму, професійним захворюванням та аварійним ситуаціям на підприємствах. Вона охоплює широкий комплекс заходів, які забезпечують безпечні умови роботи та мінімізують вплив небезпечних і шкідливих виробничих факторів на працівників. Правове регулювання охорони праці в Україні здійснюється на основі Конституції України, Кодексу законів про працю України та Закону України «Про охорону праці». Законодавство визначає права та обов'язки роботодавців і працівників, порядок організації безпечних умов праці, проведення інструктажів, медичних оглядів та навчання з питань безпеки праці. Роботодавець зобов'язаний забезпечити працівникам безпечні умови праці, належний технічний стан обладнання, засоби

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

індивідуального захисту та контроль за дотриманням вимог охорони праці. Важливою складовою охорони праці є проведення інструктажів та навчання працівників. Перед початком роботи кожен працівник повинен бути ознайомлений із правилами безпеки, можливими небезпеками на робочому місці та порядком дій у разі аварійних ситуацій. Також працівники проходять медичні огляди, особливо якщо їхня діяльність пов'язана з небезпечними або шкідливими умовами праці. Особливе значення охорона праці має на підприємствах із підвищеним рівнем небезпеки, де працівники контактують із технікою, електрообладнанням, хімічними речовинами або працюють у складних виробничих умовах. У таких випадках роботодавець повинен забезпечити не лише проведення інструктажів, а й постійний контроль за дотриманням правил безпеки. Важливу роль відіграє використання засобів індивідуального захисту, до яких належать спецодяг, каски, рукавиці, захисні окуляри, респіратори та інші засоби, що допомагають зменшити ризик травмування або професійних захворювань. Крім того, охорона праці має важливе соціальне та економічне значення. Безпечні умови праці сприяють підвищенню продуктивності працівників, зменшенню кількості нещасних випадків і втрат робочого часу через травми чи захворювання. Дотримання вимог охорони праці також позитивно впливає на репутацію підприємства та створює сприятливий психологічний клімат у колективі. Саме тому сучасні підприємства приділяють значну увагу розвитку системи управління охороною праці та впровадженню новітніх технологій безпеки. Таким чином, праця є необхідною умовою існування та розвитку суспільства, а охорона праці забезпечує безпеку та збереження здоров'я працівників у процесі трудової діяльності. Дотримання вимог охорони праці сприяє створенню безпечного робочого середовища, підвищенню продуктивності праці та зменшенню кількості виробничих травм і професійних захворювань.

5.2 Дія електричного струму на організм людини

Електричний струм є важливим фактором сучасного виробництва та побуту,

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

однак його вплив на організм людини може бути надзвичайно небезпечним. Ураження електричним струмом належить до найбільш небезпечних видів виробничих травм, оскільки навіть короточасний контакт із джерелом електроенергії може призвести до тяжких наслідків або смерті. Небезпека електричного струму полягає в тому, що людина не може заздалегідь виявити його без спеціальних приладів, а дія струму на організм часто відбувається раптово. Саме тому питання електробезпеки є важливою складовою системи охорони праці та потребує суворого дотримання правил безпеки під час роботи з електрообладнанням. Дія електричного струму на організм людини залежить від багатьох факторів, серед яких основними є сила струму, напруга, тривалість впливу, шлях проходження струму через тіло людини та фізичний стан потерпілого. Найнебезпечнішим є проходження струму через серце, легені або головний мозок, оскільки це може викликати порушення роботи життєво важливих органів. Вплив електричного струму навіть невеликої сили може спричинити судоми м'язів, сильний біль, втрату свідомості або порушення серцевого ритму. Електричний струм чинить на організм людини кілька видів дії: біологічну, теплову, хімічну та механічну. Біологічна дія проявляється у подразненні та збудженні нервової системи й м'язів, що може спричинити судоми або зупинку дихання. Теплова дія викликає опіки шкіри та внутрішніх тканин через нагрівання тканин під час проходження струму. Хімічна дія полягає у зміні складу крові та інших рідин організму, а механічна може призводити до розриву тканин або пошкодження внутрішніх органів внаслідок різкого скорочення м'язів. Ступінь ураження електричним струмом залежить також від сили струму. Струм силою до 1 мА зазвичай майже не відчувається людиною. При силі струму 10–15 мА виникають болісні відчуття та судоми м'язів, через які людина може не мати можливості самостійно відпустити провідник. Струм понад 25 мА може викликати параліч дихання, а струм силою 100 мА і більше є смертельно небезпечним, оскільки здатний спричинити фібриляцію серця та зупинку кровообігу. Велике значення має також стан навколишнього середовища. Вологість, висока температура повітря, мокрий одяг або пошкодження шкіри значно підвищують

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

ризик ураження електричним струмом. Особливо небезпечними є приміщення з підвищеною вологістю, металевими конструкціями або струмопровідними підлогами. Саме тому на виробництві використовуються спеціальні засоби захисту: діелектричні рукавиці, килимки, інструменти з ізольованими ручками та заземлення обладнання. Для запобігання ураженню електричним струмом необхідно суворо дотримуватися правил електробезпеки. Працівники повинні проходити інструктажі та навчання, перевіряти справність електрообладнання та не торкатися оголених проводів. Забороняється працювати з електроприладами мокрими руками або використовувати несправне обладнання. Важливим заходом безпеки є також відключення електроживлення перед проведенням ремонтних чи профілактичних робіт. У разі ураження людини електричним струмом необхідно негайно припинити дію струму, відключивши електроживлення або відтягнувши потерпілого за допомогою сухих ізолюючих предметів. При цьому забороняється торкатися людини голими руками, якщо вона перебуває під напругою. Після звільнення потерпілого потрібно викликати швидку медичну допомогу та, за необхідності, надати першу домедичну допомогу, зокрема штучне дихання та непрямий масаж серця. Таким чином, дія електричного струму на організм людини є дуже небезпечною та може призвести до тяжких наслідків для здоров'я і життя. Дотримання правил електробезпеки, використання засобів захисту та уважне поводження з електрообладнанням є основними умовами запобігання електротравмам у побуті та на виробництві.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено повнофункціональний вебсервіс «Interiorix», призначений для управління замовленнями у сфері створення та реалізації індивідуального інтер'єру та екстер'єру. У межах роботи було проведено аналіз існуючих рішень на ринку, визначено їх переваги та недоліки, що дозволило сформувавши технічні та функціональні вимоги до майбутнього програмного продукту.

Застосунок реалізовано на основі сучасного технологічного стеку: TypeScript, Nest.js для серверної частини, MongoDB для зберігання даних, а також React і TypeScript для реалізації зручного та інтуїтивного інтерфейсу користувача. Інтеграція усіх компонентів системи відбувалася у контейнеризованому середовищі Docker, що забезпечує простоту розгортання, масштабування та обслуговування.

Вебзастосунок підтримує ключові функції, необхідні для ефективної роботи агентства: ведення бази об'єктів з детальною інформацією, пошук та фільтрацію за параметрами, систему заявок, особистий кабінет користувача, панель адміністратора, а також інструменти для керування статусами об'єктів. Уся функціональність розроблена з урахуванням ергономічності інтерфейсу та оптимізації взаємодії між користувачами та працівниками дизайнерського агентства.

Створене рішення відповідає сучасним вимогам до вебсервісів, забезпечуючи високий рівень користувацького досвіду, а також потенціал для подальшого розвитку функціоналу.

Узагальнюючи, можна зробити висновок, що поставлені в дипломному проєкті цілі досягнуто повністю: розроблено, протестовано та задокументовано вебсервіс, який здатен не лише оптимізувати робочі процеси агентства, але й виступати конкурентоздатним цифровим продуктом на ринку. Отримані в ході проєкту знання та практичні навички є важливим внеском у професійну підготовку розробника програмного забезпечення.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Марціяш Г.Я., Слободян Р.О. Методичні вказівки до виконання дипломного проєкту для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки». Тернопіль : ВСП «ТФК ТНТУ ім. І. Пулюя», 2024, 48с.
- 2) React – офіційна документація : вебсайт. URL: <https://react.dev/> (дата звернення: 11.03.2025).
- 3) TypeScript – офіційний сайт : вебсайт. URL: <https://www.typescriptlang.org/> (дата звернення: 13.03.2025).
- 4) Docker – офіційна документація : вебсайт. URL: <https://docs.docker.com/> (дата звернення: 09.03.2025).
- 5) Material UI – компоненти для React : вебсайт. URL: <https://mui.com/> (дата звернення: 10.03.2025).
- 6) REST API Design – рекомендації Microsoft : вебсайт. URL: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design> (дата звернення: 13.03.2025).
- 7) GitHub – платформа для розробки : вебсайт. URL: <https://github.com/> (дата звернення: 07.03.2025).
- 8) Фленаган Д. *JavaScript. Повне керівництво. 7-е вид.* Київ: Видавництво «Діалектика», 2020. — 1072 с.
- 9) Фаулер М. Шаблони корпоративних застосунків. Київ: Діалектика, 2021. – 650 с.
- 10) Еванс Е. *Domain-Driven Design: Tackling Complexity in the Heart of Software.* Boston: Addison-Wesley, 2004. — 560 p.
- 11) Вест Т. Docker для розробників: створення, тестування та розгортання застосунків із використанням контейнерів. Львів: Видавництво Старого Лева, 2023. — 312 с.

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

ДОДАТКИ

Додаток А. Лістинг файлу «package.json»

```
{
  "name": "tailored-design-solutions",
  "version": "0.1.0",
  "private": true,
  "description": "Business management platform for tailored interior and
  exterior design projects.",
  "workspaces": [
    "apps/*",
    "packages/*"
  ],
  "scripts": {
    "dev": "concurrently -n backend,frontend \"npm run dev --
workspace=@tailored/backend\" \"npm run dev --
workspace=@tailored/frontend\"",
    "build": "npm run build --workspace=@tailored/shared && npm run
build --workspace=@tailored/ui && npm run build --
workspace=@tailored/backend && npm run build --
workspace=@tailored/frontend",
    "lint": "npm run lint --workspace=@tailored/backend && npm run lint
--workspace=@tailored/frontend && npm run lint --workspace=@tailored/ui &&
npm run lint --workspace=@tailored/shared",
    "format": "prettier --write
\"**/*.{ts,tsx,js,jsx,json,css,scss,md}\"",
    "test": "npm run test --workspace=@tailored/backend",
    "test:e2e": "playwright test",
    "test:e2e:ui": "playwright test --ui",
    "test:e2e:backend": "npm run test:e2e --
workspace=@tailored/backend",
    "test:all": "npm run test && npm run smoke && npm run test:e2e",
    "smoke": "node scripts/smoke-check.mjs",
```

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    "ui:smoke": "node scripts/browser-ui-check.mjs",
    "seed:report": "npm run seed:report --workspace=@tailored/backend",
    "seed": "npm run seed --workspace=@tailored/backend",
    "db:migrate": "echo \"MongoDB: no migrations – use npm run seed\"",
    "db:reset": "echo \"Reset MongoDB manually or drop database, then
npm run seed\""
  },
  "devDependencies": {
    "@playwright/test": "^1.59.1",
    "concurrently": "^9.1.2",
    "prettier": "^3.3.3",
    "typescript": "^5.5.4"
  },
  "dependencies": {
    "playwright": "^1.59.1"
  }
}

```

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

Додаток Б. Лістинг файлу «.env.example»

```
# MongoDB для `npm run dev` бекенда на вашому ПК (порт 27017 на хості).
# Скрипт start.bat / docker compose для сервісу `backend` завжди підключає
# Mongo за ім'ям сервісу `mongodb` усередині мережі Docker – це значення тут
# на контейнер бекенда не передається.
MONGODB_URI=mongodb://127.0.0.1:27017/tailored

JWT_ACCESS_SECRET=replace-with-local-access-secret
JWT_REFRESH_SECRET=replace-with-local-refresh-secret
JWT_ACCESS_TTL=24h
JWT_REFRESH_TTL=7d
FRONTEND_URL=http://localhost:3000
BACKEND_PORT=4000

# PDF receipts (company block on generated checks)
# RECEIPT_COMPANY_BRAND=Interiorix
# RECEIPT_COMPANY_LEGAL=ТОВ «Тейлорд Дизайн Солюшнс»
# RECEIPT_COMPANY_ADDRESS=м. Київ, вул. Хрещатик, 1, офіс 12
# RECEIPT_COMPANY_PHONE="+380 (44) 000-00-00"
# RECEIPT_COMPANY_EMAIL=info@tailored-design.demo
# RECEIPT_COMPANY_WEBSITE=www.tailored-design.demo
# RECEIPT_COMPANY_EDRPOU=12345678

# Vite: must include `/api` – workspace calls use
`_${VITE_API_URL}/workspace/...`
VITE_API_URL=http://localhost:4000/api

# File uploads (optional; default ./uploads relative to API cwd)
# UPLOAD_DIR=uploads
# UPLOAD_MAX_BYTES=2097152
```

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

Додаток В. Лістинг файлу «update-project-team.dto.ts»

```
import { IsOptional, IsString } from 'class-validator'
export class UpdateProjectTeamDto {
  @IsOptional()
  @IsString()
  managerStaffId?: string | null
  @IsOptional()
  @IsString()
  designerStaffId?: string | null
}
```

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

Додаток Г. Лістинг файлу «projects.module.ts»

```
import { Module } from '@nestjs/common'  
import { ProjectsController } from './projects.controller'  
import { ProjectsService } from './projects.service'  
@Module({  
  controllers: [ProjectsController],  
  providers: [ProjectsService],  
  exports: [ProjectsService],  
})  
export class ProjectsModule {}
```

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

Додаток Д. Лістинг файлу «projects.controller.ts»

```
import {
  Body,
  Controller,
  Get,
  Param,
  Patch,
  Query,
  Req,
  UseGuards,
} from '@nestjs/common'
import { ApiBearerAuth, ApiQuery, ApiTags } from '@nestjs/swagger'
import { RoleCode } from '../domain/enums'
import { AuthenticatedRequest } from '../auth/request-user'
import { JwtAuthGuard } from '../auth/jwt-auth.guard'
import { Roles } from '../auth/roles.decorator'
import { RolesGuard } from '../auth/roles.guard'
import { UpdateProjectTeamDto } from './dto/update-project-team.dto'
import { UpdateProjectStatusDto } from './dto/update-project-status.dto'
import { ProjectsService } from './projects.service'
@ApiTags('projects')
@ApiBearerAuth()
@UseGuards(JwtAuthGuard, RolesGuard)
@Controller('projects')
export class ProjectsController {
  constructor(private readonly projectsService: ProjectsService) {}
  @Get()
  @ApiQuery({ name: 'status', required: false })
  @ApiQuery({ name: 'city', required: false })
  @ApiQuery({ name: 'page', required: false })
  @ApiQuery({ name: 'perPage', required: false })
  @Roles(
    RoleCode.ADMIN,
```

					2026.КВР.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

```

        RoleCode.PROJECT_MANAGER,
        RoleCode.DESIGNER,
        RoleCode.ESTIMATOR,
        RoleCode.ACCOUNTANT
    )
    list(
        @Query('status')
        status?: string,
        @Query('city')
        city?: string,
        @Query('q')
        q?: string,
        @Query('page')
        page?: string,
        @Query('perPage')
        perPage?: string,
        @Query('from')
        from?: string,
        @Query('to')
        to?: string,
        @Query('sort')
        sort?: string
    ) {
        return this.projectsService.list({
            status,
            city,
            q: q?.trim() || undefined,
            page: page ? Number(page) : undefined,
            perPage: perPage ? Number(perPage) : undefined,
            from: from?.trim() || undefined,
            to: to?.trim() || undefined,
            sort: sort?.trim() || undefined,
        })
    }
}

```

					2026.KBP.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

```

@Get('mine')
@Roles(
    RoleCode.ADMIN,
    RoleCode.PROJECT_MANAGER,
    RoleCode.DESIGNER,
    RoleCode.ESTIMATOR
)
listMine(
    @Req()
    req: AuthenticatedRequest
) {
    return this.projectsService.listMine(req.user.id, req.user.role)
}

@Get('map')
@Roles(
    RoleCode.ADMIN,
    RoleCode.PROJECT_MANAGER,
    RoleCode.DESIGNER
)
map() {
    return this.projectsService.mapMarkers()
}

@Patch('/:id/status')
@Roles(RoleCode.ADMIN, RoleCode.PROJECT_MANAGER)
transitionStatus(
    @Param('id')
    id: string,
    @Body()
    dto: UpdateProjectStatusDto
) {
    return this.projectsService.transitionStatus(id, dto.status)
}

@Patch('/:id/team')
@Roles(RoleCode.ADMIN, RoleCode.PROJECT_MANAGER)

```

					2026.KBP.122.421.1.00.00 ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підпис	Дата		

```

updateTeam(
    @Param('id')
    id: string,
    @Body()
    dto: UpdateProjectTeamDto,
    @Req()
    req: AuthenticatedRequest
) {
    return this.projectsService.updateTeam(id, req.user.role, dto)
}
@Get('/:id/audit')
@ApiQuery({ name: 'page', required: false })
@ApiQuery({ name: 'perPage', required: false })
@Roles(
    RoleCode.ADMIN,
    RoleCode.PROJECT_MANAGER,
    RoleCode.DESIGNER,
    RoleCode.ESTIMATOR,
    RoleCode.ACCOUNTANT
)
listAudit(
    @Param('id')
    id: string,
    @Query('page')
    page?: string,
    @Query('perPage')
    perPage?: string
) {
    return this.projectsService.listAuditForProject(
        id,
        page ? Number(page) : undefined,
        perPage ? Number(perPage) : undefined
    )
}

```

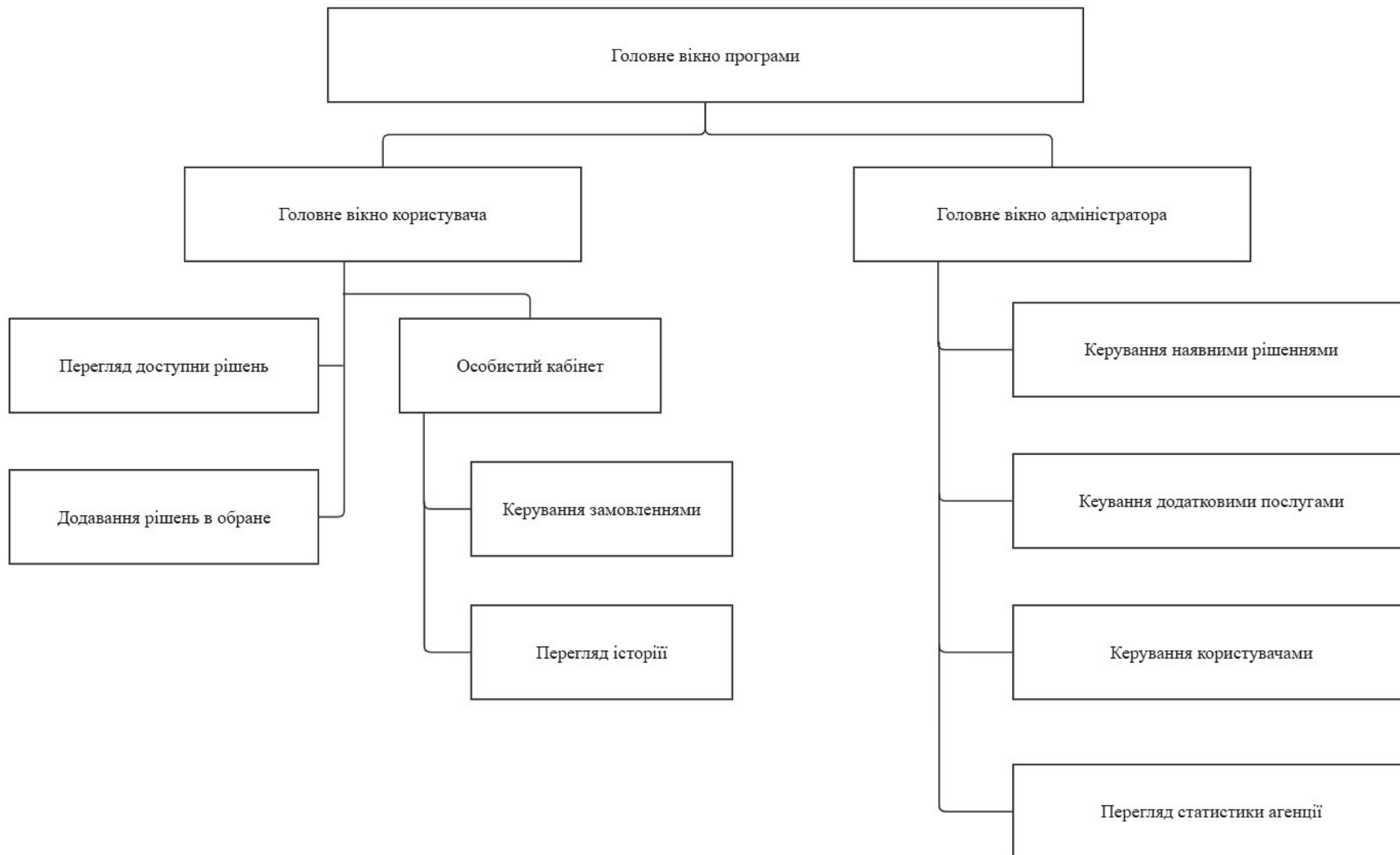
					2026.KBP.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

```

@Get('/:id')
@Roles(
    RoleCode.ADMIN,
    RoleCode.PROJECT_MANAGER,
    RoleCode.DESIGNER,
    RoleCode.ESTIMATOR,
    RoleCode.ACCOUNTANT
)
detail(
    @Param('id')
    id: string
) {
    return this.projectsService.detail(id)
}
}

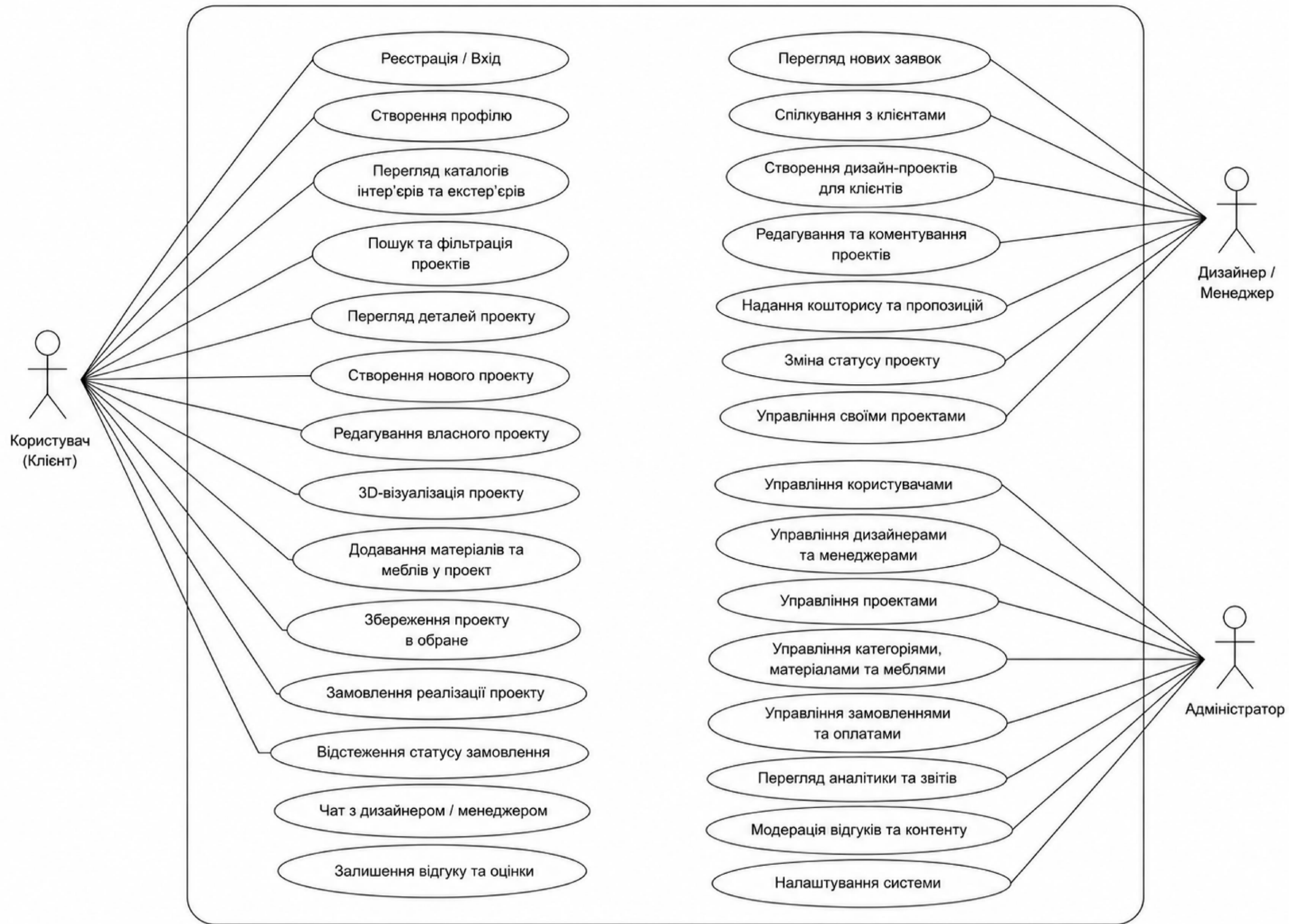
```

					2026.KBP.122.421.1.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74



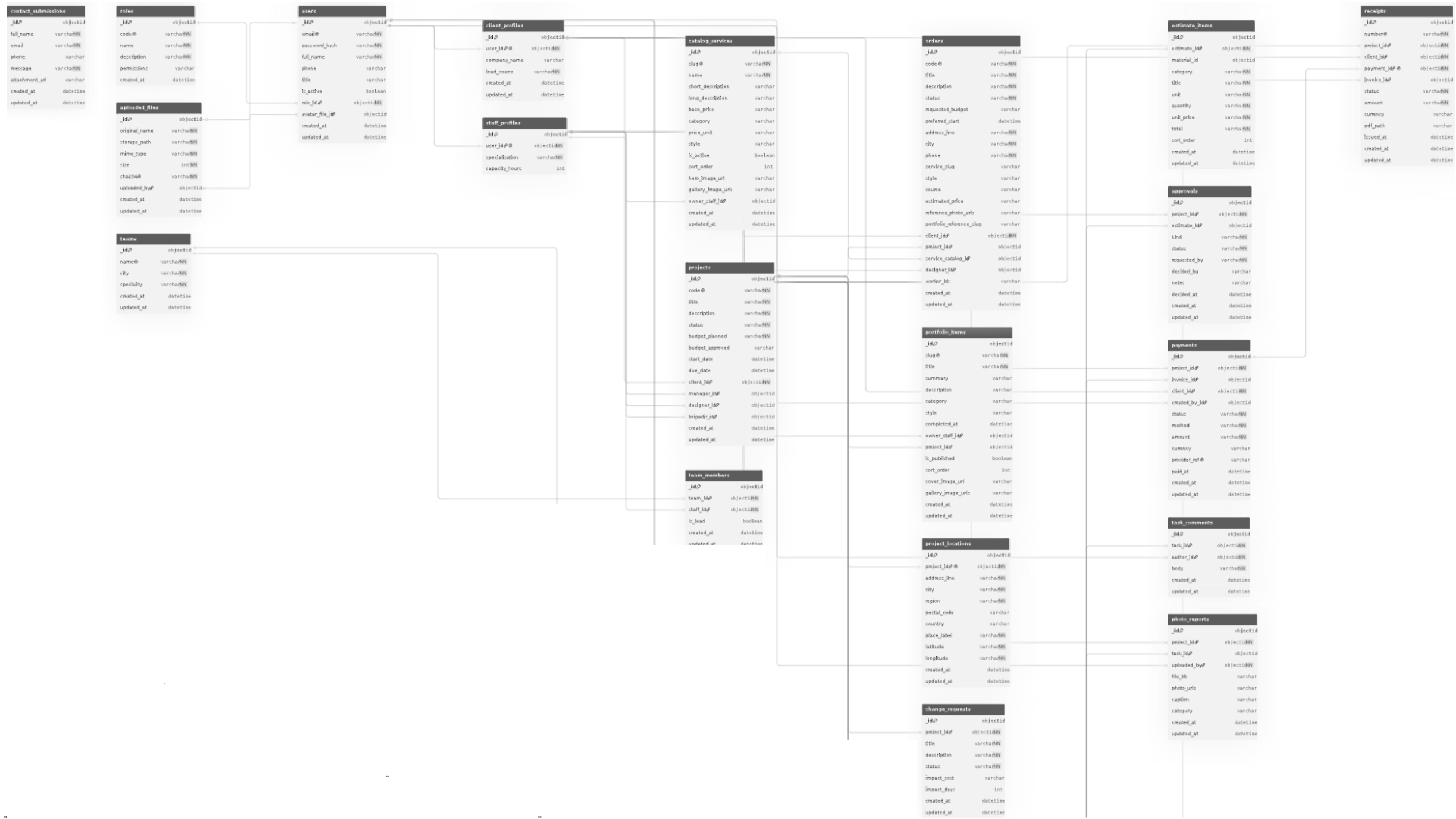
Лист № 1
Лист № 2
Лист № 3
Лист № 4
Лист № 5
Лист № 6
Лист № 7
Лист № 8
Лист № 9
Лист № 10

					2026.KBP.122.421.100.00.CC					
Зм.	Арк.	№ докум.	Лист	Дата	Розробка веб-додатку для створення та реалізації індивідуального інтер'єру по екстер'єру інтеграції. Схеми структурного веб-сайту.	Лист	Маса	Масштаб		
Розроб.		Бідацький П.В.								
Перед.		Маринка Т.Я.								
Ухвал.									Арк.шиб.	1
Розроб.									ВПТ ТФК ТНТУ КН-421	
Затв.		Примак В.А.							м. Тернопіль	
					Котлябов			Формат А1		



Лист № 1
Лист № 2
Лист № 3
Лист № 4
Лист № 5
Лист № 6
Лист № 7
Лист № 8
Лист № 9
Лист № 10
Лист № 11
Лист № 12
Лист № 13
Лист № 14
Лист № 15
Лист № 16
Лист № 17
Лист № 18
Лист № 19
Лист № 20
Лист № 21
Лист № 22
Лист № 23
Лист № 24
Лист № 25
Лист № 26
Лист № 27
Лист № 28
Лист № 29
Лист № 30
Лист № 31
Лист № 32
Лист № 33
Лист № 34
Лист № 35
Лист № 36
Лист № 37
Лист № 38
Лист № 39
Лист № 40
Лист № 41
Лист № 42
Лист № 43
Лист № 44
Лист № 45
Лист № 46
Лист № 47
Лист № 48
Лист № 49
Лист № 50

				2026.KBP.122.421.100.00 ДВ				
Зач.	Арх.	№ докум.	Лист	Лист	Розробка веб-застосунку для обробки та реалізації індивідуального інтер'єру та екстер'єру фітнес-клубу	Лист	Маса	Масштаб
Розроб.	Бабіцький П.				Додаток до проекту			
Перев.	Марчишин Т.Я.				Додаток до проекту			
І констр.						Архив	Архив	Т
Рецензент						ВПТ ТФК ТНТУ КН-421		
Начальник	Григорук В.А.					м. Тернопіль		
Затв.						Формат А1		



Лист 1/1
Лист 2/2
Лист 3/3
Лист 4/4
Лист 5/5
Лист 6/6
Лист 7/7
Лист 8/8
Лист 9/9
Лист 10/10

				2026.KBP.122.4.21.100.00 БД		
Эк	Апр	№ докум	Лист	Дата	Разработка документации для создания по реализации индивидуального интерьера по эскизам архитектора ERP-дизайна базы данных	
Резерв	Володарский П.А.				Лит	Масштаб
Титул	Маринина Г.Я.				Адапти	Т
Техзаказ					БСП ТФК ТНТУ - КН-421	
Решения					м. Тернополь	
Начерт	Григорук В.А.				Копировать	
Зам					Формат А1	

Таблиця техніко-економічних показників

№	Показник	Одиниці вимірювання	Значення
1	Мова програмування	–	TypeScript
2	Технології клієнтської частини	–	HTML, CSS, React
3	Технології серверної частини	–	Nest.js
4	Середовище розробки	–	Web Storm
5	Загальний розмір програми	Мб	190
6	База даних	–	MongoBD
7	Міграція бази даних	–	Flyway
8	Собівартість проекту	грн	72 977
9	Плановий прибуток	грн	40 867
10	Ціна	грн	113 844
11	Чиста теперішня вартість	грн	28 654
12	Термін окупності	рік	2,1

				2026.KBP.122.4.21.100.00 ТБ		
Мен	Лист	№ докум	Підп	Дата	Лист	Масштаб
Розроб	Білодський ІВ					
Проб	Марціши Г.Я.					
І.контр						
І.контр	Піришмак В.А.					
Білод						
				Розробка вебзастосунку для створення та реалізації індивідуального інтер'єру та екстер'єру «Інтелігент» Таблиця техніко-економічних показників		
				Лист	Листів	7
				ВП ТФК ТНТУ КН-421 м. Тернопіль		
				Копіювати Формат А1		