

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

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

«Вебсервіс для бронювання житла»

Виконала студентка групи ПЗ-22-1
ПИЛИПЕНКО Єлизавета Володимирівна

Керівник роботи:
ЛОКТИКОВА Тамара Миколаївна

Рецензент:
БАКАЛЮК Тетяна Анатоліївна

Житомир – 2026

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

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна ВАКАЛЮК

«28» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ПИЛИПЕНКО Єлизавета Володимирівна**

Керівник роботи: **ЛОКТИКОВА Тамара Миколаївна**

Тема роботи: **«Вебсервіс для бронювання житла»**,

затверджена Наказом закладу вищої освіти від **«27» лютого 2026 р., №92/с**

Вихідні дані для роботи: **об'єктом дослідження є процеси пошуку, вибору, бронювання та керування об'єктами житлової нерухомості в онлайн-середовищі, а предметом дослідження є методи, алгоритми та програмні засоби побудови веборієнтованої системи управління бронюванням житла.**

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Локтікова Т. М.	02.03.2026р.	02.03.2026р.
2	Локтікова Т. М.	02.03.2026р.	02.03.2026р.
3	Локтікова Т. М.	02.03.2026р.	02.03.2026р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебсервісу для бронювання житла та пояснювальної записки. Пояснювальна записка містить 92 сторінки текстової частини, 42 рисунки, 11 таблиць. Список використаних джерел налічує 32 та займає 3 сторінки. Робота включає 1 додаток. Загальний обсяг роботи складає 102 сторінки.

Метою роботи є проектування та програмна реалізація вебсервісу для бронювання житла, який забезпечує пошук доступних об'єктів, створення й супровід бронювань, керування оголошеннями, обмін повідомленнями, модерацію даних і розмежування прав доступу користувачів.

У роботі виконано аналіз предметної області, спроектовано структуру вебсервісу, базу даних, алгоритми функціонування та реалізовано програмне забезпечення для бронювання житла.

Серверну частину вебсервісу реалізовано мовою PHP із використанням фреймворку Laravel. Клієнтську частину побудовано на основі Vue.js 3, Inertia.js і Tailwind CSS. Для зберігання даних використано PostgreSQL, для збирання клієнтських ресурсів — Vite, а для організації контейнеризованого середовища розробки — Docker і Laravel Sail.

КЛЮЧОВІ СЛОВА: ВЕБСЕРВІС, БРОНЮВАННЯ ЖИТЛА, LARAVEL, VUE.JS, INERTIA.JS, POSTGRESQL, DOCKER.

					ІПЗ.КР.Б-121-26-ІПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	Вебсервіс для бронювання житла			Літ.	Арк.	Аркушів	
Розроб.		Пилипенко Є.В.									
Керівник		Локтікова Т.М.							3	102	
								Житомирська політехніка, група ІПЗ-22-1			
Н. контр.		Тичина О.П.									
Затвердив		Вакалюк Т.А.									

ABSTRACT

The bachelor's qualification work consists of a web service for housing booking and an explanatory note. The explanatory note contains 92 pages of text, 42 figures, and 11 tables. The list of references includes 32 sources and occupies 3 pages. The work contains 1 appendix. The total length of the work is 102 pages.

The aim of the work is to design and implement a housing booking web service that provides searching for available properties, creating and managing bookings, listing management, message exchange, data moderation, and user access control.

The work includes an analysis of the subject area, the design of the web service structure, database, and operating algorithms, as well as the implementation of software for housing booking.

The server side of the web service was implemented in PHP using the Laravel framework. The client side was developed using Vue.js 3, Inertia.js, and Tailwind CSS. PostgreSQL was used for data storage, Vite for building client-side resources, and Docker together with Laravel Sail for organizing the containerized development environment.

KEYWORDS: WEB SERVICE, HOUSING BOOKING, LARAVEL, VUE.JS, INERTIA.JS, POSTGRESQL, DOCKER.

		Пилипенко Є.В.			ІІЗ.КР.Б-121-26-ІІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБСЕРВІСУ	9
1.1. Постановка задачі	9
1.2. Аналіз аналогів програмного продукту	12
1.3. Вибір архітектури вебсервісу	18
1.4. Обґрунтування вибору інструментальних засобів реалізації	24
Висновки до розділу 1	28
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБСЕРВІСУ	29
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи	29
2.2. Розробка бази даних системи.....	37
2.3. Проєктування та реалізація алгоритмів роботи системи	45
2.4. Проєктування бізнес-правил вебсервісу.....	51
2.5. Реалізація функціоналу вебсервісу	55
Висновки до розділу 2	59
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБСЕРВІСОМ	61
3.1. Порядок встановлення та налаштування параметрів системи	61
3.2. Структура інтерфейсу та порядок взаємодії з вебсервісом	64
3.3. Тестування роботи вебсервісу	82
Висновки до розділу 3	87
ВИСНОВКИ	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	89
ДОДАТКИ	92

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних.

ІПЗ – інженерія програмного забезпечення.

ПЗ – програмне забезпечення.

СУБД – система управління базами даних.

API – прикладний програмний інтерфейс (Application Programming Interface).

CSS – каскадні таблиці стилів (Cascading Style Sheets).

ER – модель «сутність–зв’язок» (Entity–Relationship).

HTML – мова гіпертекстової розмітки (HyperText Markup Language).

HTTP – протокол передавання гіпертексту (HyperText Transfer Protocol).

ORM – технологія об’єктно-реляційного відображення даних (Object-Relational Mapping).

PDF – формат переносного документа (Portable Document Format).

PHP – мова програмування PHP (PHP: Hypertext Preprocessor).

REST – архітектурний стиль взаємодії вебсервісів (Representational State Transfer).

SQL – мова структурованих запитів (Structured Query Language).

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми. Розвиток цифрових технологій супроводжується впровадженням веборієнтованих інформаційних систем у сферу надання послуг. Одним із актуальних напрямів є автоматизація пошуку та короткострокового бронювання житла. Спеціалізовані вебсервіси забезпечують централізоване зберігання інформації про об'єкти нерухомості, пошук доступних пропозицій, контроль бронювань і взаємодію між орендарями, власниками житла та адміністраторами платформи.

Підвищення ефективності й надійності бронювання є необхідним, оскільки ручне опрацювання заявок і використання різних каналів комунікації можуть спричиняти помилки у виборі дат, дублювання бронювань, несвоєчасне оновлення доступності та неточності розрахунку вартості проживання. Єдиний вебсервіс дає змогу автоматизувати основні операції, забезпечити цілісність даних і спростити взаємодію учасників процесу.

Метою випускної кваліфікаційної роботи бакалавра є проєктування та програмна реалізація вебсервісу для бронювання житла, який забезпечує пошук доступних об'єктів, створення й супровід бронювань, керування оголошеннями, обмін повідомленнями, модерацію даних і розмежування прав доступу користувачів.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

- 1) проаналізувати предметну область і наявні вебсервіси короткострокового бронювання житла;
- 2) сформулювати функціональні, нефункціональні та системні вимоги до програмного продукту;
- 3) визначити ролі користувачів і основні сценарії їхньої взаємодії із системою;
- 4) спроєктувати архітектуру вебсервісу, структуру програмних сутностей і реляційну базу даних;
- 5) розробити алгоритми перевірки доступності житла, запобігання перетину часових інтервалів і розрахунку вартості проживання;

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

6) реалізувати серверну та клієнтську частини вебсервісу, механізми автентифікації, авторизації, валідації даних і розмежування прав доступу;

7) реалізувати функції пошуку, бронювання, керування оголошеннями, модерації, обміну повідомленнями, додавання об'єктів до обраного та створення відгуків;

8) провести функціональне тестування основних сценаріїв роботи системи.

Об'єктом дослідження є процеси пошуку, вибору, бронювання та керування об'єктами житлової нерухомості в онлайн-середовищі.

Предметом дослідження є методи, алгоритми та програмні засоби побудови веборієнтованої системи управління бронюванням житла.

Практичне значення отриманих результатів полягає у створенні працездатного вебсервісу, який автоматизує основні операції з оголошеннями та бронюваннями, забезпечує контроль доступності житла й підтримує взаємодію між орендарями, власниками та адміністраторами.

За результатами виконання випускної кваліфікаційної роботи опубліковано тези:

1) Пилипенко Є.В., Локтікова Т.М., Кушнір Н.О. Сучасний стан і тенденції розвитку сервісів для бронювання житла: тези VIII Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 02-03 грудня 2025 р. Житомир: Житомирська політехніка, 2025. С. 59-60 [1];

2) Пилипенко Є.В., Локтікова Т.М., Кушнір Н.О. Дослідження принципів побудови та проєктування вебсервісу для бронювання житла: тези XVI Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології - 2026», м. Житомир, 02-03 квітня 2026 р. Житомир: Житомирська політехніка, 2026. С. 55-56 [2].

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБСЕРВІСУ

1.1. Постановка задачі

Процес короткострокового бронювання житла охоплює пошук доступних об'єктів, вибір періоду проживання, визначення кількості гостей, розрахунок вартості та оформлення бронювання. У разі ручного опрацювання заявок або використання різних каналів комунікації можуть виникати помилки під час вибору дат, дублювання бронювань, несвоєчасне оновлення інформації про доступність житла та неточності у визначенні вартості проживання.

Для автоматизації зазначених процесів необхідно розробити вебсервіс, який забезпечуватиме централізоване зберігання та опрацювання даних про користувачів, об'єкти житла, бронювання, відгуки й повідомлення. Система повинна підтримувати пошук житла, перевірку його доступності, створення і супровід бронювань, керування оголошеннями, модерацію даних та взаємодію між учасниками платформи.

У вебсервісі мають бути передбачені три основні категорії користувачів: неавторизований користувач, авторизований користувач та адміністратор. Авторизований користувач залежно від виконуваних дій може виступати як орендар або власник житла.

Неавторизованому користувачеві необхідно надати можливість переглядати активні оголошення, здійснювати пошук і фільтрацію житла та ознайомлюватися з характеристиками вибраного об'єкта. Для створення бронювання, додавання житла до обраного, обміну повідомленнями та виконання інших персоналізованих дій користувач повинен пройти реєстрацію й автентифікацію.

Авторизований користувач у ролі орендаря повинен мати можливість:

- виконувати пошук житла за заданими параметрами;
- переглядати опис, фотографії, місцезнаходження, вартість та умови проживання;

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

- обирати дати заїзду і виїзду та вказувати кількість гостей;
- отримувати автоматично розраховану вартість проживання;
- створювати бронювання за умови доступності об'єкта;
- переглядати історію та скасовувати незавершені бронювання;
- додавати житло до списку обраного;
- обмінюватися повідомленнями та залишати відгуки після завершення проживання.

Авторизований користувач у ролі власника житла повинен мати можливість створювати, редагувати та видаляти власні оголошення. Під час додавання об'єкта необхідно зазначати його назву, опис, тип, адресу, місто, координати, ціну за ніч, допустиму кількість гостей і доплату за додаткового гостя. До оголошення повинно бути додано не менше трьох фотографій.

Перед публікацією нове оголошення має пройти перевірку адміністратором. За результатами модерації воно може бути активовано або деактивовано із зазначенням причини. Адміністратор також повинен мати можливість керувати користувачами та бронюваннями, переглядати статистичні показники й підтримувати комунікацію з користувачами системи.

Однією з основних задач вебсервісу є запобігання одночасному бронюванню одного об'єкта на часові інтервали, що перетинаються. Перед створенням нового бронювання система повинна перевіряти наявні активні записи для вибраного житла. Нове бронювання може бути створене лише за умови коректності введених дат і відсутності конфлікту з уже зарезервованими періодами.

Вартість проживання повинна визначатися автоматично з урахуванням кількості ночей, ціни за одну ніч і доплати за додаткових гостей. Після підтвердження бронювання система має зберігати загальну суму оплати та суму, що нараховується власнику житла. Власнику нараховується 90 % загальної вартості бронювання, а комісія платформи становить 10 % і визначається як різниця між загальною сумою оплати та сумою, нарахованою власнику.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

Для збереження історії даних скасовані бронювання не повинні фізично видалятися. У таких випадках змінюється їхній статус, що дозволяє надалі переглядати відповідну інформацію. Скасування завершених бронювань не допускається, а можливість залишення відгуку надається лише після завершення періоду проживання.

Вебсервіс повинен забезпечувати розмежування прав доступу відповідно до ролі користувача, перевірку введених даних і захист персональної інформації. Інтерфейс системи має бути зрозумілим, адаптивним і придатним для використання на пристроях із різними розмірами екрана.

Окрему групу вимог до вебсервісу становлять вимоги до вхідних і вихідних даних. До вхідних даних належать реєстраційні відомості користувачів, параметри пошуку, характеристики об'єктів житла, фотографії, дати заїзду та виїзду, кількість гостей, тексти повідомлень і відгуків. Під час отримання таких даних система повинна перевіряти їх тип, повноту, допустимі межі та відповідність установленим правилам. Некоректні значення не повинні зберігатися в базі даних, а користувачеві має надаватися зрозуміле повідомлення про причину помилки.

Вихідними даними системи є переліки доступних об'єктів житла, відомості про оголошення, результати розрахунку вартості, підтвердження створення або скасування бронювання, повідомлення між користувачами, статистичні показники та результати адміністративної модерації. Подання інформації повинно залежати від ролі користувача. Відкриті відомості про активні оголошення можуть переглядати всі відвідувачі, тоді як персональні дані, історія бронювань, повідомлення та адміністративні функції мають бути доступними лише після автентифікації та перевірки відповідних прав.

До нефункціональних вимог належать надійність, безпека, зручність використання, адаптивність і можливість подальшого супроводу вебсервісу. Система повинна зберігати узгодженість пов'язаних даних, коректно опрацьовувати помилкові запити та не допускати виконання операцій користувачами, які не мають необхідних прав. Паролі мають зберігатися у

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

вигляді криптографічних хешів, а операції створення, редагування та видалення даних повинні виконуватися через захищені маршрути з перевіркою автентифікації.

Користувацький інтерфейс повинен однаково коректно відображатися на персональних комп'ютерах, планшетах і мобільних пристроях. Елементи керування мають бути логічно згрупованими, а результати виконання операцій — супроводжуватися повідомленнями про успіх або помилку. Для спрощення подальшого розвитку програмний код необхідно розділити відповідно до функціонального призначення його компонентів, а доступ до даних організувати через моделі та визначені між ними зв'язки.

Критеріями завершеності розроблення є можливість пройти основні сценарії роботи системи: зареєструватися та автентифікуватися, знайти житло, переглянути його характеристики, створити бронювання на вільний період, скасувати незавершене бронювання, додати оголошення, пройти його модерацію, обмінятися повідомленнями та залишити відгук після завершеного проживання. Адміністратор повинен мати змогу контролювати користувачів, оголошення і бронювання, а також переглядати узагальнену інформацію про роботу вебсервісу.

Результатом розв'язання поставленої задачі має бути працездатний вебсервіс для бронювання житла, який автоматизує основні процеси короткострокової оренди, забезпечує цілісність і актуальність даних, запобігає конфліктам бронювання та підтримує взаємодію між орендарями, власниками житла й адміністраторами в межах єдиної інформаційної системи.

1.2. Аналіз аналогів програмного продукту

Для визначення функціональних можливостей, які мають бути передбачені у вебсервісі бронювання житла, необхідно проаналізувати існуючі програмні продукти аналогічного призначення. Аналіз аналогів дозволяє визначити типовий набір функцій систем даного класу, оцінити підходи до організації пошуку житла, бронювання, комунікації між користувачами та адміністрування

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

даних. У межах підрозділу розглянуто вебсервіси Doba.ua, Dobovo.com та Airbnb, які відрізняються масштабом використання, функціональними можливостями та способом організації взаємодії між орендарями й власниками житла.

Першим аналогом є сервіс Doba.ua, який спеціалізується на пошуку та розміщенні оголошень про подовову оренду житла в Україні [3]. Платформа надає користувачеві можливість переглядати доступні об'єкти нерухомості, ознайомлюватися з описом житла, фотографіями, місцем розташування, вартістю проживання та контактною інформацією власника. Інтерфейс сервісу орієнтований на швидкий пошук пропозицій за містом і параметрами проживання, а зовнішній вигляд головної сторінки Doba.ua представлено на рис. 1.1.

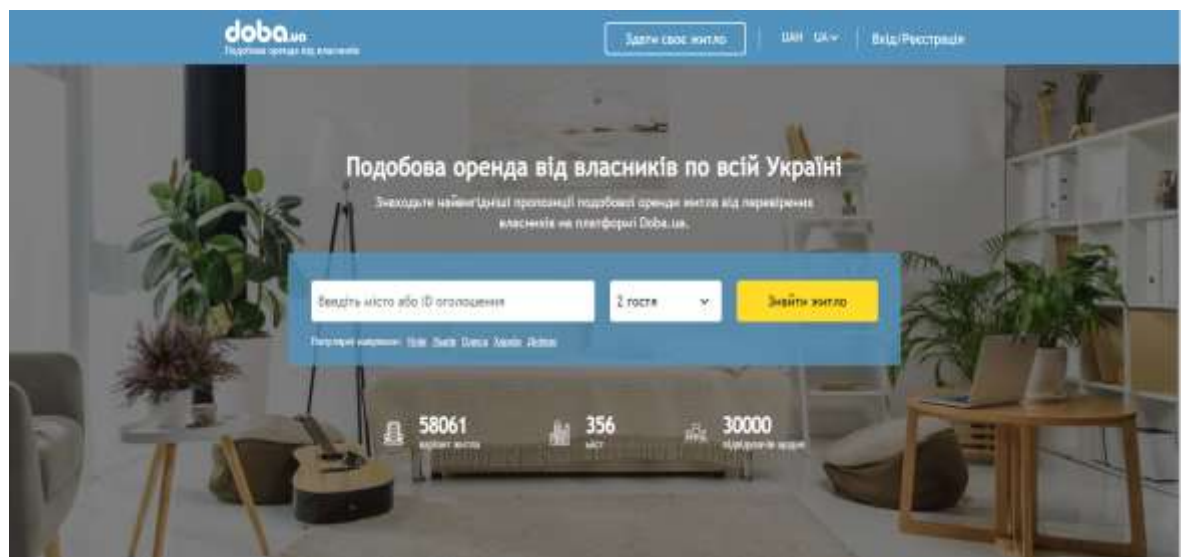


Рис. 1.1. Головна сторінка вебсервісу Doba.ua

Сервіс Doba.ua забезпечує базову інформаційну підтримку процесу оренди житла: користувач може переглядати оголошення, порівнювати пропозиції за вартістю, розташуванням і характеристиками, а також отримувати відомості для подальшого зв'язку з власником. Перевагою такого підходу є простота використання та орієнтація на український ринок подовової оренди. Водночас сервіс більшою мірою виконує роль каталогу оголошень, тому частина взаємодії між орендарем і власником може здійснюватися поза межами платформи. Це знижує рівень автоматизації процесу бронювання та може ускладнювати контроль актуальності доступних дат.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Другим аналогом є Dobovo.com — український вебсервіс, призначений для пошуку та бронювання житла подового [4]. Платформа дозволяє користувачеві обрати місто, дати проживання, кількість гостей і переглянути доступні варіанти розміщення. На сторінках об'єктів подається опис житла, фотографії, інформація про умови проживання, розташування та вартість. Особливістю Dobovo.com є орієнтація саме на швидкий вибір житла за основними параметрами проживання. Користувач одразу бачить доступні пропозиції та може порівнювати їх за розташуванням, вартістю й умовами розміщення. Такий підхід є корисним для розроблюваного вебсервісу, оскільки підтверджує доцільність реалізації пошуку житла за локацією та бюджетом. Структура головної сторінки сервісу Dobovo.com відображає орієнтацію на швидке оформлення бронювання через вебінтерфейс. Головна сторінка сервісу Dobovo.com представлена на рис. 1.2.

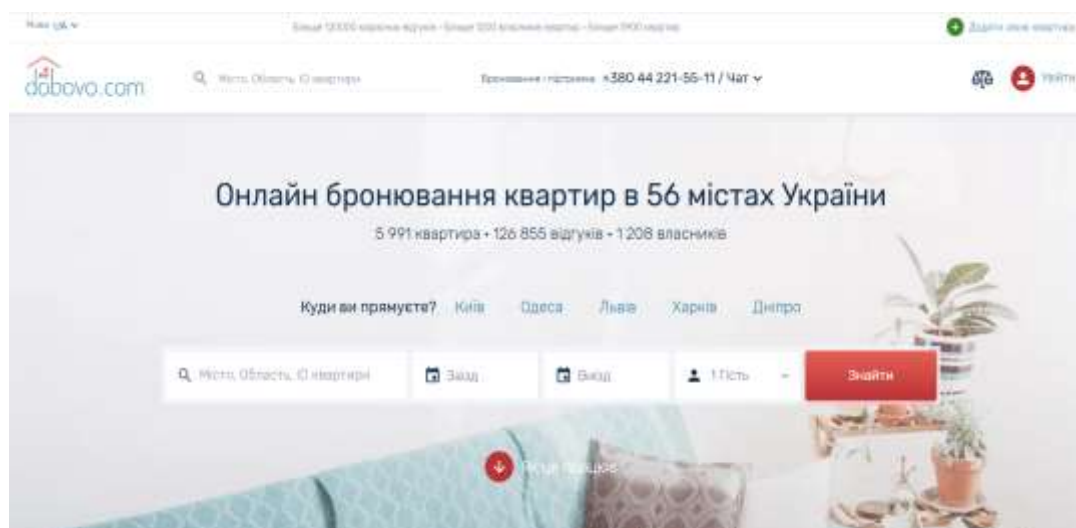


Рис. 1.2. Головна сторінка вебсервісу Dobovo.com

Dobovo.com має більш виражену спрямованість на автоматизацію процесу бронювання порівняно з каталогами оголошень. Користувач може не лише знайти житло, а й перейти до оформлення бронювання безпосередньо через вебсервіс. До переваг системи належать зручне подання результатів пошуку, наявність фільтрації, структурований опис об'єктів і можливість перегляду житла за основними параметрами. Недоліком є те, що сервіс орієнтований на широкий комерційний ринок, тому його функціональність може бути надмірною для локального або навчального вебсервісу. Крім того, ефективність роботи

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

таких платформ залежить від своєчасного оновлення інформації про доступність об'єктів власниками житла.

Третім аналогом є Airbnb — міжнародна платформа для пошуку та бронювання житла, яка реалізує повний цикл взаємодії між користувачами та власниками об'єктів розміщення [5]. Сервіс надає можливість виконувати пошук за місцем призначення, датами, кількістю гостей, типом житла та іншими параметрами. Користувач може переглядати фотографії, описи, рейтинги, відгуки, правила проживання та доступність об'єкта. Інтерфейс Airbnb є прикладом функціонально розвиненої системи бронювання житла з великою кількістю сервісних можливостей. Головна сторінка Airbnb показана на рис. 1.3.

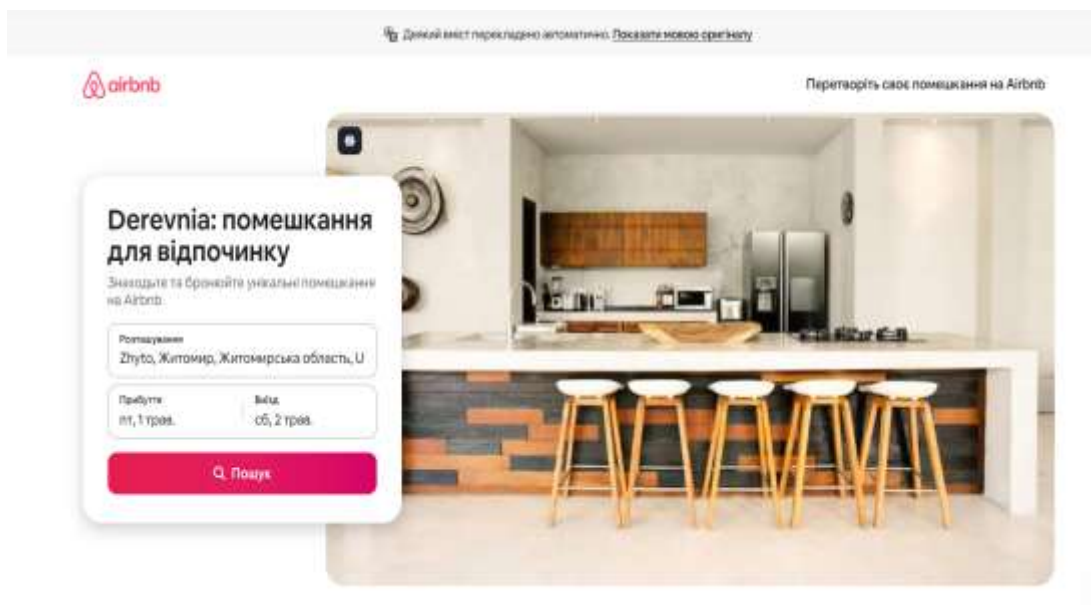


Рис. 1.3. Головна сторінка вебсервісу Airbnb

Airbnb забезпечує не лише пошук і бронювання житла, а й внутрішню комунікацію між користувачами, підтримку відгуків, систему довіри, керування календарем доступності та обробку платежів. Перевагою сервісу є високий рівень автоматизації процесів і значна кількість інструментів для орендарів та власників житла. Водночас платформа має складну систему правил, комісій, перевірок і обмежень, що є виправданим для міжнародного сервісу, але може бути надлишковим для розроблюваної системи. Тому під час проектування власного вебсервісу доцільно врахувати базові принципи роботи Airbnb, але реалізувати їх у простішій і більш керованій формі.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Для систематизації результатів аналізу виконано порівняння функціональних можливостей розглянутих аналогів (табл. 1.1).

Таблиця 1.1

Порівняльна характеристика систем управління бронюванням

Функціональна можливість	Doba.ua	Dobovo.com	Airbnb
Автоматизоване онлайн-бронювання	+	+	+
Адаптивність інтерфейсу	+	+	+
Система сповіщень користувачів	-	+	+
Контроль актуальності календаря доступності	+	+	+
Внутрішня система повідомлень	-	-	+
Фінансова статистика для власника	-	-	+
Адміністративна статистика системи	-	-	+

Підтримка світлої та темної теми	-	-	-
--	---	---	---

Проведене порівняння показує, що всі розглянуті системи забезпечують базовий функціонал, необхідний для пошуку та бронювання житла. До такого функціоналу належать онлайн-бронювання, адаптивний інтерфейс, підтримка календаря доступності та інструменти для взаємодії між користувачами і власниками житла.

Вебсервіс Doba.ua орієнтований переважно на ринок подовогої оренди житла в Україні та забезпечує користувачам можливість швидкого пошуку й бронювання об'єктів нерухомості. Основною перевагою сервісу є простота використання та орієнтація на локальний ринок. Водночас система має обмежені можливості щодо адміністрування та аналізу фінансових показників.

Dobovo.com також орієнтований на ринок подовогої оренди житла та надає користувачам більш широкий набір сервісних функцій, зокрема систему сповіщень і зручні засоби керування бронюваннями. Проте функції статистичного аналізу та розширеного адміністрування залишаються обмеженими.

Airbnb є найбільш функціонально розвиненою платформою серед розглянутих аналогів. Сервіс підтримує внутрішню систему повідомлень, фінансові інструменти для власників житла, адміністративні механізми керування та розвинену систему бронювання. Разом із тим через значний масштаб платформи її структура та правила використання є більш складними порівняно з локальними сервісами.

На відміну від розглянутих аналогів, розроблюваний вебсервіс поєднує базові можливості систем бронювання житла з додатковими функціями адміністрування та контролю. У системі реалізовано внутрішню взаємодію між користувачами, фінансову статистику для власників житла, адміністративну

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

статистику, підтримку світлої та темної тем оформлення інтерфейсу, а також механізми контролю стану оголошень і бронювань.

Таким чином, результати аналізу аналогів були використані для формування функціональних вимог до розроблюваного вебсервісу та визначення можливостей, які мають бути реалізовані під час проєктування і розроблення системи.

1.3. Вибір архітектури вебсервісу

Для реалізації вебсервісу бронювання житла обрано клієнт-серверну архітектуру. За такого підходу користувач взаємодіє із системою через веббраузер, який надсилає HTTP-запити до сервера. Сервер приймає запит, виконує перевірку отриманих даних, опрацьовує відповідну операцію та повертає результат клієнтській частині [6]. Централізоване виконання основних операцій на сервері дає змогу забезпечити узгоджене опрацювання бронювань, контроль прав доступу та збереження даних.

Архітектуру вебсервісу побудовано за багаторівневим принципом, відповідно до якого окремі компоненти системи мають визначені обов'язки та взаємодіють між собою через встановлені механізми [7]. У розробленій системі виділено рівень представлення, прикладний рівень, рівень доступу до даних та інфраструктурний рівень. Такий поділ забезпечує логічне розмежування функцій системи, спрощує її супровід і створює можливість подальшого розширення.

Рівень представлення відповідає за відображення інформації та взаємодію користувача з вебсервісом. Клієнтську частину реалізовано з використанням Vue.js 3. Цей фреймворк забезпечує компонентний підхід до побудови інтерфейсу та дає змогу створювати динамічні сторінки, форми, фільтри, модальні вікна, календарі й адміністративні панелі [8].

У структурі клієнтської частини окремими Vue-компонентами реалізовано заголовок і нижню частину сторінки, картки оголошень, фільтри категорій, карту, форми введення даних, компоненти профілю користувача й елементи адміністративної панелі. Повторне використання компонентів зменшує

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

дублювання програмного коду та забезпечує однакову поведінку елементів на різних сторінках системи.

Для оформлення користувацького інтерфейсу застосовано Tailwind CSS. Його використання дає змогу задавати стилі безпосередньо в структурі компонентів, створювати адаптивне оформлення та підтримувати єдиний вигляд елементів вебсервісу [9]. У системі передбачено коректне відображення сторінок на персональних комп'ютерах і мобільних пристроях, а також перемикання між світлою та темною темами.

Прикладний рівень системи реалізовано засобами фреймворку Laravel. Він відповідає за маршрутизацію запитів, автентифікацію користувачів, перевірку вхідних даних, виконання бізнес-правил і формування відповідей [10]. На цьому рівні опрацьовуються операції пошуку житла, створення і скасування бронювань, керування оголошеннями, додавання об'єктів до обраного, створення відгуків, надсилання повідомлень і виконання адміністративних дій.

Організація серверного коду відповідає архітектурному шаблону MVC, який передбачає розподіл програмного забезпечення на моделі, представлення та контролери [11]. Моделі використовуються для подання основних сутностей предметної області та взаємодії з базою даних. Контролери приймають запити, виконують валідацію, перевіряють установлені правила та передають результат до клієнтської частини. Представлення формуються Vue-компонентами, які відображають отримані від сервера дані.

У серверній частині передбачено окремі контролери для роботи з користувачами, об'єктами житла, бронюваннями, повідомленнями, відгуками та обраним. Адміністративні функції винесено в окрему групу контролерів, що відповідають за керування користувачами, оголошеннями, бронюваннями та статистичними показниками. Такий поділ дає змогу згрупувати методи відповідно до їхнього функціонального призначення.

Для зв'язку між Laravel і Vue.js використовується Inertia.js [12]. Він дозволяє поєднати серверну маршрутизацію Laravel із клієнтськими Vue-компонентами без розроблення окремого REST API для кожної сторінки.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

Контролер визначає компонент, який потрібно відобразити, і передає йому необхідний набір даних. Під час переходу між сторінками Inertia.js оновлює лише необхідну частину інтерфейсу без повного перезавантаження вебсторінки.

Таким чином, розроблений вебсервіс є монолітним вебзастосунком із динамічним клієнтським інтерфейсом. Серверна та клієнтська частини розміщуються в одному проєкті й використовують спільну систему маршрутизації. Такий підхід спрощує автентифікацію, передавання результатів валідації, формування повідомлень про помилки та підтримку узгодженого стану між сервером і користувацьким інтерфейсом.

Рівень доступу до даних реалізовано за допомогою ORM Eloquent, що входить до складу Laravel. Моделі системи відповідають таблицям бази даних і містять опис зв'язків між сутностями. До основних моделей належать User, Property, PropertyImage, Booking, Review, Message та Favorite. Через моделі контролери отримують, додають, змінюють і видаляють записи без безпосереднього формування SQL-запитів для кожної операції.

Для постійного зберігання інформації обрано реляційну систему керування базами даних PostgreSQL [13]. Її використання відповідає структурі предметної області, оскільки сутності вебсервісу мають чітко визначені зв'язки. Користувач може створювати декілька оголошень і бронювань, об'єкт житла може містити декілька фотографій та відгуків, а повідомлення пов'язуються з відправником і отримувачем.

Цілісність даних забезпечується первинними та зовнішніми ключами, типами даних, обов'язковими полями й іншими обмеженнями структури таблиць [14]. Наприклад, бронювання повинно належати наявному користувачеві та конкретному об'єкту житла, а відгук повинен містити посилання на автора й оголошення. У разі видалення пов'язаних записів поведінка системи визначається правилами, установленними в міграціях бази даних.

Розподіл функцій між компонентами дає змогу впорядкувати структуру вебсервісу, забезпечити окреме виконання завдань кожного рівня та спростити

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

внесення подальших змін. Узагальнену архітектуру вебсервісу наведено на рис. 1.4.

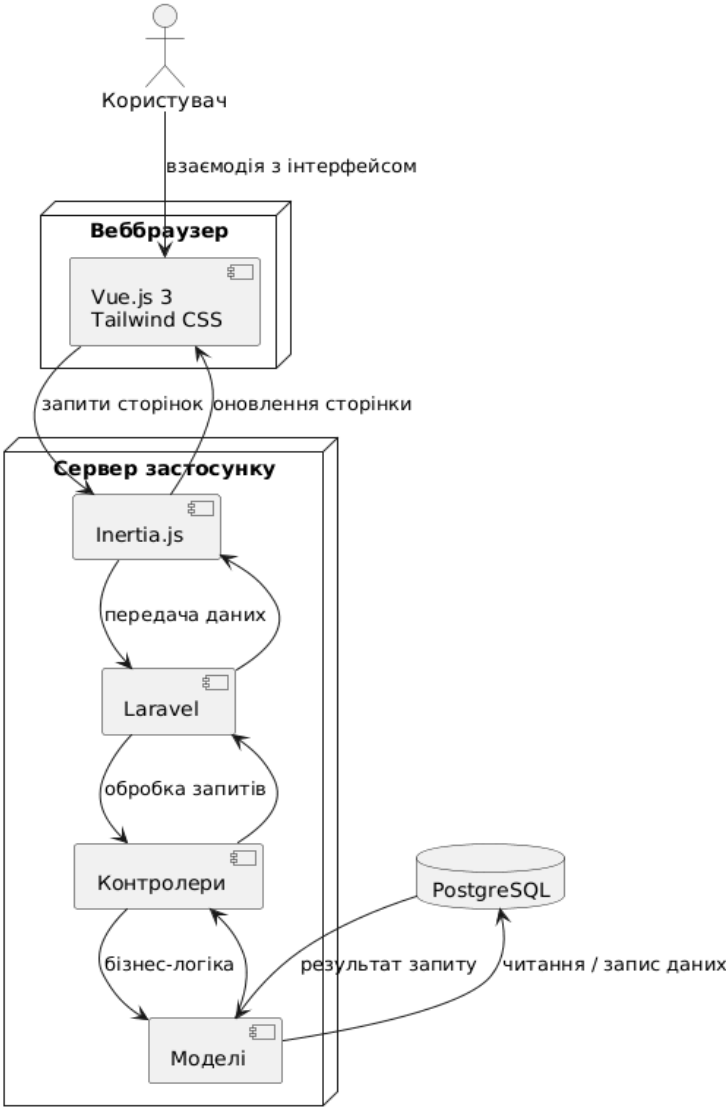


Рис. 1.4. Узагальнена архітектура вебсервісу

Взаємодія компонентів вебсервісу починається з надсилання користувачем запиту через веббраузер. Запит надходить до серверної частини Laravel, де маршрутизатор визначає відповідний контролер. Контролер перевіряє отримані дані, виконує необхідну бізнес-логіку, звертається до моделей і бази даних PostgreSQL, після чого передає результат до Vue-компонента через Inertia.js. Клієнтська частина відображає отримані дані у вигляді сторінки або оновленого елемента інтерфейсу.

Важливою складовою обраної архітектури є виконання основних бізнес-правил на серверному рівні. Клієнтська частина може попередньо перевіряти

заповнення полів форми та повідомляти користувача про некоректні значення, однак остаточна перевірка даних виконується сервером. Такий підхід дає змогу запобігти збереженню некоректної інформації та виконанню операцій із порушенням установлених правил.

Під час створення бронювання система перевіряє правильність дат заїзду та виїзду, кількість гостей і наявність активних бронювань для вибраного житла. Нове бронювання створюється лише за відсутності перетину часових інтервалів. При цьому суміжні бронювання дозволяються, тобто дата виїзду одного користувача може збігатися з датою заїзду іншого.

Розрахунок вартості проживання також виконується на серверному рівні. Система визначає кількість ночей, базову вартість проживання та доплату за додаткових гостей. Значення, передане клієнтською частиною, не використовується як остаточне, оскільки перед створенням бронювання сервер повторно виконує розрахунок загальної суми.

Обрана архітектура забезпечує розмежування прав доступу відповідно до категорії користувача. Неавторизований користувач може переглядати відкриті сторінки, здійснювати пошук і фільтрацію житла. Авторизований користувач може створювати бронювання, додавати об'єкти до обраного, залишати відгуки після завершення проживання, обмінюватися повідомленнями та керувати власними оголошеннями. Адміністратор має доступ до керування користувачами, оголошеннями, бронюваннями та статистичними показниками роботи вебсервісу.

Окремою складовою архітектури є підсистема обміну повідомленнями. Повідомлення зберігаються в базі даних і пов'язуються з відправником та отримувачем. Після створення нового повідомлення система формує подію, яка передається отримувачеві в реальному часі. Доступ до приватного каналу надається лише авторизованому користувачеві, якому призначене повідомлення.

Для розробленого вебсервісу обрано монолітну архітектуру, за якої серверна та клієнтська частини розміщуються в одному проєкті та використовують спільну систему маршрутизації. Такий підхід відповідає обсягу

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

розробленої системи, спрощує передавання даних, перевірку прав доступу та відображення помилок валідації. Водночас програмний код логічно розділено на моделі, контролери, компоненти та окремі функціональні модулі.

Використання мікросервісної архітектури для розробленої системи є недоцільним, оскільки поділ функцій бронювання, оголошень, повідомлень та автентифікації на незалежні сервіси ускладнив би їх розгортання, налаштування взаємодії та підтримку цілісності даних. Монолітний підхід дає змогу зберегти простішу структуру проєкту та водночас забезпечити розмежування функцій між його компонентами.

Для організації середовища розробки використовується Docker. Контейнеризація дає змогу запускати компоненти вебсервісу в ізольованому та попередньо налаштованому середовищі, зменшує залежність від параметрів операційної системи та спрощує розгортання проєкту на іншому комп'ютері [15].

Керування контейнеризованим середовищем виконується за допомогою Laravel Sail [16]. Він забезпечує запуск контейнерів вебзастосунку, бази даних PostgreSQL, Redis і сервера передавання подій у реальному часі, а також виконання команд Laravel усередині підготовленого середовища.

Для збирання клієнтських ресурсів використовується Vite [17]. Він опрацьовує Vue-компоненти, JavaScript-файли та стилі, забезпечує швидке оновлення внесених змін під час розробки й формує оптимізовані ресурси для запуску вебсервісу.

Отже, для вебсервісу обрано клієнт-серверну багаторівневу архітектуру монолітного застосунку. Рівень представлення реалізовано за допомогою Vue.js 3 і Tailwind CSS, прикладний рівень — засобами Laravel, а взаємодію між ними забезпечує Inertia.js. Для зберігання даних використовується PostgreSQL, для збирання клієнтських ресурсів — Vite, а для організації середовища розробки — Docker і Laravel Sail. Обрана архітектура забезпечує розмежування відповідальності між компонентами, виконання бізнес-правил на серверному рівні та можливість подальшого розвитку вебсервісу.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

1.4. Обґрунтування вибору інструментальних засобів реалізації

Для реалізації вебсервісу бронювання житла необхідно обрати інструментальні засоби, які забезпечують стабільну роботу серверної частини, створення зручного користувацького інтерфейсу, надійне зберігання даних і можливість подальшого розвитку програмного продукту. Під час вибору технологій ураховано їхню сумісність, наявність документації, поширеність, підтримку сучасних підходів до веброзробки та відповідність функціональним вимогам і характеристикам якості програмного забезпечення [18, 19].

До складу вебсервісу входять серверна та клієнтська частини, база даних і допоміжні сервіси. Серверна частина відповідає за опрацювання запитів, перевірку вхідних даних, виконання бізнес-правил і розмежування прав доступу. Клієнтська частина забезпечує взаємодію користувача із системою, а база даних використовується для зберігання інформації про користувачів, об'єкти житла, бронювання, повідомлення, відгуки та обрані оголошення.

Для реалізації серверної частини обрано мову програмування PHP [20, 21]. Вона широко використовується для створення вебзастосунків і підтримує роботу з HTTP-запитами, сесіями, файлами та базами даних. Виконання програмного коду на сервері дає змогу централізовано перевіряти отримані значення, виконувати основні операції системи та приховувати внутрішню логіку вебсервісу від користувача. У розробленому проєкті використовується PHP версії 8.3 або новішої.

Використання PHP без додаткового фреймворку потребувало б окремої реалізації маршрутизації, автентифікації, валідації, доступу до бази даних і захисту запитів. Тому серверну частину побудовано на основі фреймворку Laravel [10, 11]. Він надає готову структуру проєкту та засоби для роботи з маршрутами, контролерами, моделями, міграціями, сесіями й повідомленнями.

Засобами Laravel реалізовано пошук житла, створення і скасування бронювань, керування оголошеннями, обмін повідомленнями, додавання об'єктів до обраного, створення відгуків та виконання адміністративних

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

операцій. Перевірка вхідних даних здійснюється перед їх збереженням, а доступ до захищених функцій контролюється за допомогою проміжного програмного забезпечення, політик і перевірки ролі користувача.

Для створення клієнтської частини обрано Vue.js 3 [8, 22]. Компонентний підхід цього фреймворку дає змогу поділяти інтерфейс на окремі повторно використовувані елементи. У проєкті у вигляді Vue-компонентів реалізовано картки оголошень, форми, фільтри, модальні вікна, календар, карту, сторінки профілю та елементи адміністративної панелі.

Vue.js забезпечує оновлення інтерфейсу відповідно до зміни даних без повного перезавантаження сторінки. Це використовується під час фільтрації оголошень, вибору дат проживання, визначення кількості гостей, відображення вартості бронювання, керування фотографіями та перегляду статистичних даних. Повторне використання компонентів зменшує дублювання програмного коду й забезпечує однакову поведінку елементів на різних сторінках.

Для зв'язку між Laravel і Vue.js використовується Inertia.js [12]. Він дає змогу поєднати серверну маршрутизацію Laravel із клієнтськими Vue-компонентами без створення окремого REST API для кожної сторінки. Контролер визначає компонент, який потрібно відобразити, і передає йому необхідні дані, тоді як автентифікація, валідація та перевірка прав залишаються на стороні Laravel.

Застосування Inertia.js є доцільним, оскільки серверна та клієнтська частини вебсервісу розміщені в одному проєкті. Це спрощує передавання даних, відображення помилок валідації та підтримання узгодженого стану користувацького інтерфейсу.

Для оформлення інтерфейсу застосовано Tailwind CSS [9]. Він містить набір допоміжних класів для визначення розмірів, відступів, шрифтів, меж, розташування та інших параметрів елементів. Це дає змогу підтримувати єдиний вигляд компонентів без створення значної кількості окремих CSS-правил.

Tailwind CSS використовується для оформлення сторінок пошуку, карток житла, форм бронювання, профілю, чату й адміністративної панелі. Під час

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

розроблення враховано принципи адаптивного вебдизайну, тому сторінки коректно відображаються на пристроях із різними розмірами екрана [23]. У системі також передбачено перемикання між світлою та темною темами.

Для постійного зберігання інформації обрано реляційну СУБД PostgreSQL [13, 14]. Її використання відповідає структурі предметної області, оскільки сутності вебсервісу мають чітко визначені зв'язки. Користувач може створювати декілька оголошень і бронювань, об'єкт житла може містити декілька фотографій та відгуків, а повідомлення пов'язуються з відправником і отримувачем.

PostgreSQL підтримує первинні та зовнішні ключі, обмеження, індекси, транзакції й різні типи даних. Це дає змогу забезпечити цілісність інформації та не допускати створення записів, які посилаються на відсутні сутності. У базі даних зберігаються відомості про користувачів, житло, фотографії, бронювання, повідомлення, відгуки та обрані оголошення.

Доступ до таблиць здійснюється через ORM Eloquent, що входить до складу Laravel. Моделі User, Property, PropertyImage, Booking, Review, Message та Favorite відповідають основним сутностям системи й містять опис зв'язків між ними. Використання Eloquent спрощує отримання пов'язаних даних і зменшує необхідність безпосереднього формування SQL-запитів у контролерах.

Для виконання окремих функцій вебсервісу використано допоміжні клієнтські бібліотеки. Компонент v-calendar забезпечує вибір дат проживання та відображення зайнятих періодів. Для роботи з картою й координатами житла застосовано Leaflet. Графіки адміністративної статистики формуються за допомогою Chart.js, а створення звітів у форматі PDF виконується засобами jsPDF і jspdf-autotable.

Для асинхронних HTTP-запитів використовується Axios. Обмін повідомленнями в реальному часі реалізовано за допомогою Laravel Reverb і Laravel Echo. Пакет pusher-js застосовується на клієнтській стороні для встановлення WebSocket-з'єднання з Reverb, а не як окремий зовнішній сервіс.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

Для збирання клієнтських ресурсів застосовується Vite [17]. Він опрацьовує JavaScript-файли, Vue-компоненти та стилі, забезпечує їх підключення до Laravel-застосунку й швидке оновлення під час розроблення. Під час підготовки системи до розгортання Vite формує оптимізовану збірку ресурсів.

Для організації середовища розробки використовується Docker [15]. Він дає змогу запускати компоненти системи в окремих ізольованих контейнерах із визначеними параметрами. Завдяки цьому зменшується залежність проєкту від конфігурації операційної системи та спрощується його запуск на іншому комп'ютері.

Керування контейнеризованим середовищем здійснюється за допомогою Laravel Sail [16]. У конфігурації проєкту передбачено сервіси для Laravel-застосунку, PostgreSQL, Redis і Laravel Reverb. Laravel Sail використовується для запуску контейнерів, виконання міграцій, установлення залежностей і запуску команд Artisan. Redis входить до середовища як допоміжний сервіс, який може застосовуватися для кешування або опрацювання черг.

Написання та редагування програмного коду здійснюється в середовищі Visual Studio Code. Воно підтримує роботу з PHP, JavaScript, Vue-компонентами, стилями, конфігурацією Docker та міграціями бази даних. Для контролю версій використовується система Git [24], яка дає змогу зберігати історію змін, працювати з гілками та порівнювати версії файлів.

Керування залежностями серверної частини здійснюється за допомогою Composer [25]. Він завантажує PHP-пакети, визначені в конфігурації проєкту, і забезпечує їх автоматичне підключення. Установлення клієнтських залежностей виконується за допомогою менеджера пакетів npm [26], який працює в середовищі Node.js [27]. За допомогою npm встановлюються Vue.js, Inertia.js, Tailwind CSS, Vite та допоміжні бібліотеки.

Використання зазначених засобів визначає вимоги до програмного й апаратного середовища. Для розроблення вебсервісу потрібна 64-розрядна операційна система Windows, Linux або macOS із підтримкою Docker,

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

установлені Git і Composer, не менше 4 ГБ оперативної пам'яті та достатній обсяг вільного місця. Для одночасного запуску всіх компонентів рекомендовано 8 ГБ оперативної пам'яті та SSD-накопичувач, тоді як кінцевому користувачеві достатньо сучасного веббраузера з підтримкою JavaScript і доступом до мережі.

Отже, обрані засоби забезпечують реалізацію серверної та клієнтської частин, зберігання даних, збирання ресурсів і запуск вебсервісу в контейнеризованому середовищі та відповідають функціональним вимогам системи.

Висновки до розділу 1

У першому розділі проаналізовано предметну область короткострокового бронювання житла та визначено основні проблеми, які виникають під час ручного опрацювання заявок: дублювання бронювань, несвоєчасне оновлення доступності об'єктів, помилки під час вибору дат і неточності в розрахунку вартості проживання. Обґрунтовано доцільність створення вебсервісу, який автоматизує ці процеси та забезпечує централізоване зберігання даних.

Під час аналізу аналогів розглянуто сервіси Doba.ua, Dobovo.com та Airbnb. Визначено їхні основні переваги й обмеження, що дало змогу сформулювати перелік функцій для розроблюваної системи: пошук і фільтрацію житла, перегляд детальної інформації про об'єкти, перевірку доступності дат, створення бронювань, систему повідомлень, модерацію оголошень і розмежування прав доступу користувачів.

Для реалізації вебсервісу обґрунтовано вибір клієнт-серверної архітектури та інструментальних засобів розробки. Серверну частину доцільно реалізувати на основі Laravel, клієнтську частину — з використанням Vue.js 3, Inertia.js і Tailwind CSS, а для зберігання даних використати PostgreSQL. Також визначено доцільність застосування Vite, Docker і Laravel Sail для організації середовища розробки та збирання клієнтських ресурсів.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБСЕРВІСУ

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

Проектування вебсервісу передбачає визначення основних категорій користувачів і сценаріїв їхньої взаємодії із системою. На цьому етапі необхідно встановити, які функції доступні кожному типу користувача та які дії виконуються в межах основних бізнес-процесів бронювання житла.

У вебсервісі передбачено три основні категорії користувачів: неавторизований користувач, авторизований користувач та адміністратор. Авторизований користувач може виконувати дії орендаря або власника житла залежно від поточного сценарію роботи із системою.

Неавторизований користувач має доступ до відкритої частини вебсервісу. Він може переглядати активні оголошення, виконувати пошук, фільтрацію та сортування житла, відкривати сторінку конкретного об'єкта, переглядати фотографії, опис, ціну, рейтинг, відгуки та розташування житла на карті. Для виконання персоналізованих дій користувач повинен пройти реєстрацію або автентифікацію.

Авторизований користувач має доступ до функцій бронювання житла, додавання об'єктів до обраного, створення відгуків після завершення проживання, скасування незавершених бронювань та обміну повідомленнями. Цей самий користувач може також виконувати функції власника житла: створювати й редагувати власні оголошення, додавати фотографії, подавати оголошення на модерацію, переглядати бронювання власних об'єктів і фінансові показники.

Орендар і власник житла не є окремими типами облікових записів, а визначаються діями, які виконує авторизований користувач у межах поточного сценарію роботи із системою. Тому на діаграмі варіантів використання відповідні функції об'єднано в межах одного актора — «Авторизований користувач».

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

Адміністратор забезпечує контроль роботи вебсервісу. Він має доступ до адміністративної частини системи, де може керувати користувачами, оголошеннями, бронюваннями, повідомленнями та статистичними показниками. Адміністратор виконує модерацію нових оголошень, активує або деактивує їх із зазначенням причини, переглядає аналітичні дані та відстежує підозрілу активність користувачів.

Для забезпечення повного циклу роботи з вебсервісом кожна категорія користувачів отримує набір функціональних можливостей відповідно до своїх повноважень. Неавторизовані користувачі працюють лише з відкритою частиною системи, тоді як авторизовані користувачі отримують доступ до створення бронювань, керування обраним, залишення відгуків і обміну внутрішніми повідомленнями. Авторизовані користувачі, які розміщують власні оголошення, додатково можуть керувати ними, переглядати бронювання власних об'єктів і відповідні фінансові показники. Адміністратор виконує контрольні та управлінські функції.

Під час побудови діаграми варіантів використання враховано, що авторизований користувач може виконувати різні функції залежно від обраного сценарію взаємодії із системою. Він має можливість бронювати житло, керувати списком обраного, обмінюватися повідомленнями, залишати відгуки, а також створювати й редагувати власні оголошення. Таким чином, функції орендаря та власника житла подано в межах одного актора — «Авторизований користувач», що відповідає логіці роботи вебсервісу.

Адміністратор, на відміну від звичайних користувачів, не бере участі в процесі бронювання як орендар або власник житла, а виконує функції контролю та супроводу системи. До його повноважень належать модерація оголошень, керування користувачами, контроль бронювань, перегляд статистичних і аналітичних даних, а також опрацювання звернень через внутрішні чати. Такий поділ повноважень дає змогу чітко відокремити користувацькі сценарії від адміністративних функцій.

Основні варіанти використання вебсервісу наведено на рис. 2.1.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		30



Рис. 2.1. Діаграма варіантів використання вебсервісу

Діаграма варіантів використання відображає розподіл функцій між основними категоріями користувачів. Неавторизований користувач працює лише з відкритою частиною системи. Авторизований користувач отримує доступ до бронювання, обраного, відгуків, повідомлень і створення власних оголошень. У межах вебсервісу авторизований користувач може виконувати дії як орендаря, так і власника житла, тому ці сценарії подано в межах одного актора — авторизованого користувача. Адміністратор має розширені можливості

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

керування системою, зокрема модерацію оголошень, зміну статусів бронювань, керування користувачами та перегляд аналітики.

Опис основних варіантів використання наведено в табл. 2.1.

Таблиця 2.1

Опис основних варіантів використання вебсервісу

Варіант використання	Основний актор	Короткий опис
Реєстрація та автентифікація	Неавторизований користувач	Користувач створює обліковий запис або входить до системи для доступу до персоналізованих функцій вебсервісу
Пошук, фільтрація та перегляд житла	Неавторизований користувач, авторизований користувач	Користувач задає місто, період проживання, кількість гостей, бюджет та інші параметри, після чого переглядає активні оголошення, фотографії, опис, вартість, рейтинг, відгуки й розташування житла.
Створення та підтвердження бронювання	Авторизований користувач	Користувач обирає житло, дати проживання та кількість гостей. Система перевіряє доступність об'єкта, розраховує вартість проживання й фінансові показники та після підтвердження створює бронювання.
Перегляд і скасування бронювань	Авторизований користувач	Користувач переглядає власні бронювання та може скасувати бронювання, якщо період проживання ще не завершився.
Керування обраним	Авторизований користувач	Користувач додає об'єкти житла до списку обраного, переглядає збережені об'єкти та видаляє їх зі списку.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

Продовження табл. 2.1

Обмін повідомленнями	Авторизований користувач	Користувач надсилає та отримує повідомлення, відповідає на них і переглядає статус їх прочитання.
Звернення до підтримки	Авторизований користувач	Користувач створює звернення до адміністратора та отримує відповідь через внутрішню систему повідомлень.
Залишення відгуку після завершення проживання	Авторизований користувач	Користувач залишає оцінку й текстовий коментар про об'єкт житла після завершення відповідного періоду проживання.
Створення та керування власними оголошеннями	Авторизований користувач	Користувач створює, редагує та видаляє власні оголошення, зазначає характеристики житла, додає фотографії й передає оголошення на модерацію.
Перегляд бронювань і фінансових показників власних об'єктів	Авторизований користувач	Користувач переглядає бронювання власних об'єктів, загальну вартість бронювань і суми, що нараховуються власнику житла.
Керування користувачами	Адміністратор	Адміністратор переглядає та редагує відомості про користувачів, а також блокує або розблоковує їхні облікові записи.
Модерація та керування оголошеннями	Адміністратор	Адміністратор перевіряє нові оголошення, активує або деактивує їх, зазначає причину відхилення та контролює опубліковані пропозиції.

Продовження табл. 2.1

Керування бронюваннями	Адміністратор	Адміністратор переглядає бронювання користувачів, контролює їхній стан і за потреби змінює статус.
Опрацювання звернень і чатів	Адміністратор	Адміністратор переглядає звернення користувачів, відповідає на повідомлення та забезпечує комунікацію зі службою підтримки.
Перегляд аналітики та формування PDF-звітів	Адміністратор	Адміністратор переглядає статистику користувачів, оголошень, бронювань і фінансових показників, аналізує карту об'єктів та підозрілу активність, а також формує звіти у форматі PDF.

Для взаємодії з таблицями БД у Laravel використовується ORM-підхід, за якого основні сутності системи подаються у вигляді моделей. Це дає змогу працювати з даними на рівні програмних об'єктів, описувати зв'язки між таблицями в коді та спростувати взаємодію між контролерами, моделями й базою даних.

Об'єктно-орієнтована структура вебсервісу реалізується за допомогою моделей Laravel, які відображають основні сутності предметної області [10,11]. До ключових моделей системи належать User, Property, Booking, PropertyImage, Message, Review та Favorite. Зазначені моделі забезпечують збереження інформації про користувачів, житло, бронювання, фотографії, повідомлення, відгуки й обрані об'єкти.

Кожна модель відповідає окремій логічній сутності вебсервісу та має зв'язки з іншими моделями відповідно до процесів роботи системи. Наприклад, користувач може мати декілька оголошень і бронювань, оголошення може містити декілька фотографій, а бронювання пов'язується одночасно з

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

користувачем і конкретним об'єктом житла. Таке подання структури системи дає змогу узгодити програмні моделі з майбутньою структурою бази даних.

Використання моделей також спрощує подальшу реалізацію основних функцій вебсервісу. Через зв'язки між моделями система може отримувати фотографії конкретного житла, бронювання певного користувача, повідомлення між учасниками та відгуки до обраного об'єкта. Це забезпечує узгоджену роботу програмної логіки та структури даних. Завдяки цьому зменшується дублювання коду під час опрацювання однакових типів даних у різних частинах системи. Крім того, така структура полегшує подальше розширення вебсервісу новими функціями без суттєвої зміни основної логіки роботи.

Для відображення основних моделей вебсервісу та зв'язків між ними побудовано діаграму класів засобами PlantUML [28], яку наведено на рис. 2.2.

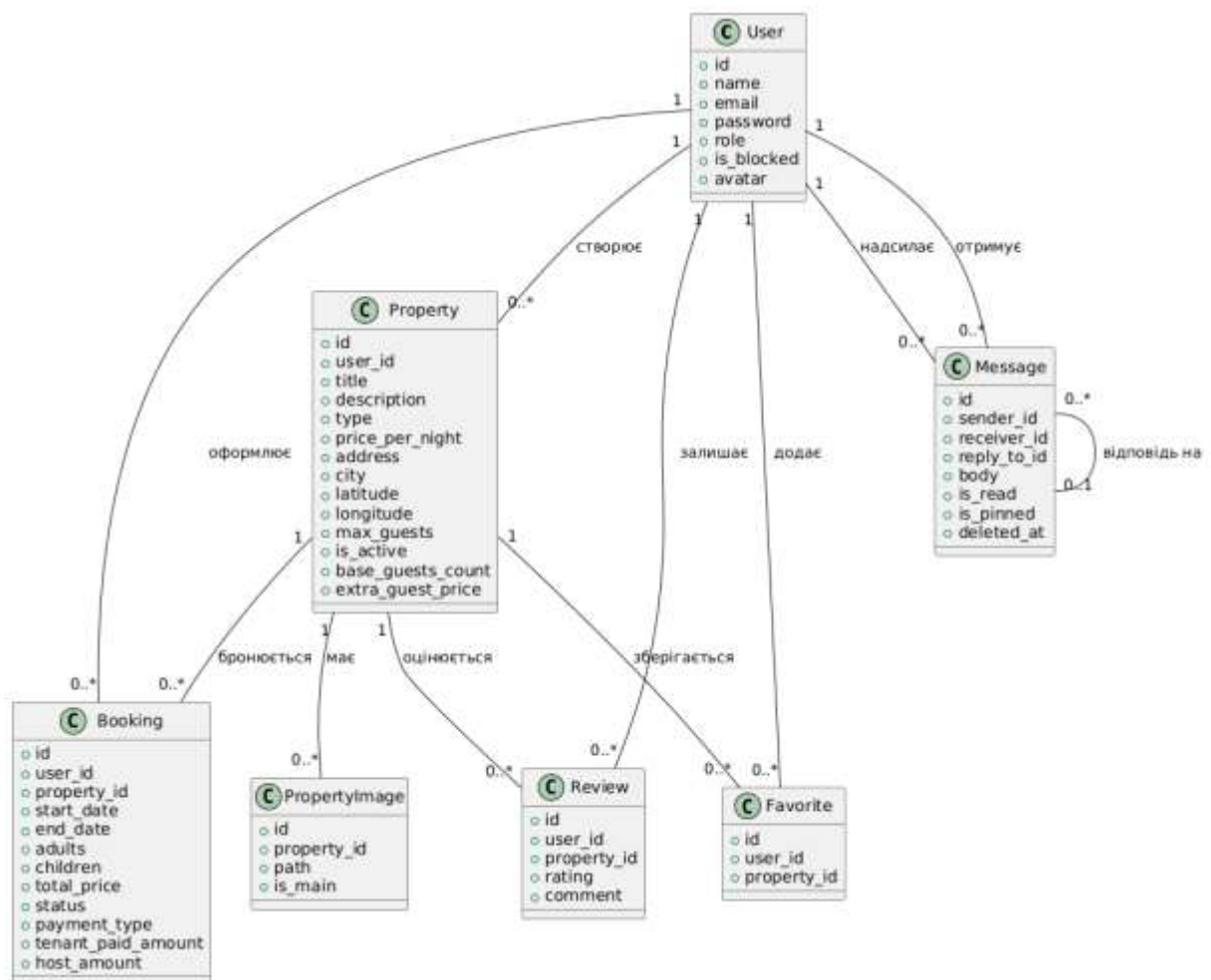


Рис. 2.2. Діаграма класів вебсервісу

Модель User описує користувача системи. Вона містить ім'я, електронну пошту, пароль, роль, ознаку блокування та аватар. Роль користувача використовується для розмежування доступу між звичайними користувачами й адміністратором. Користувач може мати власні оголошення, бронювання, відгуки, обрані об'єкти, а також надіслані та отримані повідомлення.

Модель Property описує об'єкт житла, який розміщується у вебсервісі. Вона містить назву, опис, тип житла, ціну за ніч, адресу, місто, географічні координати, максимальну кількість гостей, базову кількість гостей, доплату за додаткового гостя та ознаку активності. Кожен об'єкт житла належить певному користувачу-власнику та може мати декілька фотографій, бронювань, відгуків і записів у списку обраного.

Модель PropertyImage використовується для зберігання фотографій житла. Вона містить шлях до зображення та ознаку головної фотографії. Такий підхід дозволяє пов'язувати з одним об'єктом житла декілька фотографій і визначати основне зображення для відображення у списку оголошень.

Модель Booking відповідає за бронювання житла. Вона пов'язує користувача з конкретним об'єктом, містить дати заїзду та виїзду, кількість дорослих і дітей, загальну вартість, статус бронювання, тип оплати, суму, сплачену користувачем, і суму, що нараховується власнику. Через цю модель система контролює доступність житла на визначений період і формує фінансові показники.

Модель Message забезпечує внутрішню комунікацію в системі. Вона містить ідентифікатори відправника й отримувача, текст повідомлення, посилання на повідомлення, на яке надається відповідь, статус прочитання, ознаку закріплення та підтримку м'якого видалення. Завдяки цій моделі реалізовано зв'язок між користувачами, власниками житла та адміністрацією.

Модель Review використовується для збереження відгуків. Вона містить ідентифікатор користувача, ідентифікатор житла, оцінку та текстовий коментар. Відгук пов'язується з конкретним об'єктом і користувачем, а можливість його створення надається лише після завершення проживання.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

Модель Favorite відповідає за додавання житла до обраного. Вона реалізує зв'язок між користувачем і об'єктом житла, який було збережено для повторного перегляду. Це підвищує зручність користування вебсервісом під час вибору житла.

Таким чином, у підрозділі визначено основні ролі користувачів, варіанти використання вебсервісу та об'єктно-орієнтовану структуру системи. Запропонована структура відповідає реальній реалізації програмного продукту та створює основу для подальшого проектування бази даних і алгоритмів роботи системи.

2.2. Розробка бази даних системи

Для ідентифікації кожного запису в таблицях використовуються первинні ключі, а зв'язки між сутностями реалізуються за допомогою зовнішніх ключів. Зокрема, таблиці оголошень, бронювань, відгуків і повідомлень містять посилання на відповідних користувачів, а фотографії, бронювання, відгуки та записи обраного пов'язуються з конкретними об'єктами житла. Такий підхід забезпечує цілісність даних і запобігає появі записів, які не мають відповідних пов'язаних сутностей.

Зв'язок між користувачами й об'єктами, доданими до обраного, реалізовано через проміжну таблицю, що дає змогу одному користувачеві зберігати декілька оголошень, а одному оголошенню одночасно перебувати у списках обраного різних користувачів. Структуру таблиць розділено відповідно до функціонального призначення даних, що зменшує їх дублювання, спрощує виконання запитів і забезпечує можливість подальшого розширення вебсервісу.

Окреме зберігання фотографій дає змогу пов'язувати з одним оголошенням декілька зображень і визначати головну фотографію об'єкта. Дані про бронювання містять інформацію про користувача, вибране житло, період проживання, кількість гостей, загальну вартість і поточний статус бронювання. Таблиця повідомлень забезпечує збереження історії комунікації між відправником і отримувачем, а також дає змогу відстежувати стан прочитання

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

повідомлень. Передбачена структура бази даних підтримує основні бізнес-процеси вебсервісу та забезпечує узгоджене опрацювання пов'язаних даних.

Фізична структура бази даних вебсервісу подана у вигляді ER-діаграми на рис. 2.3.

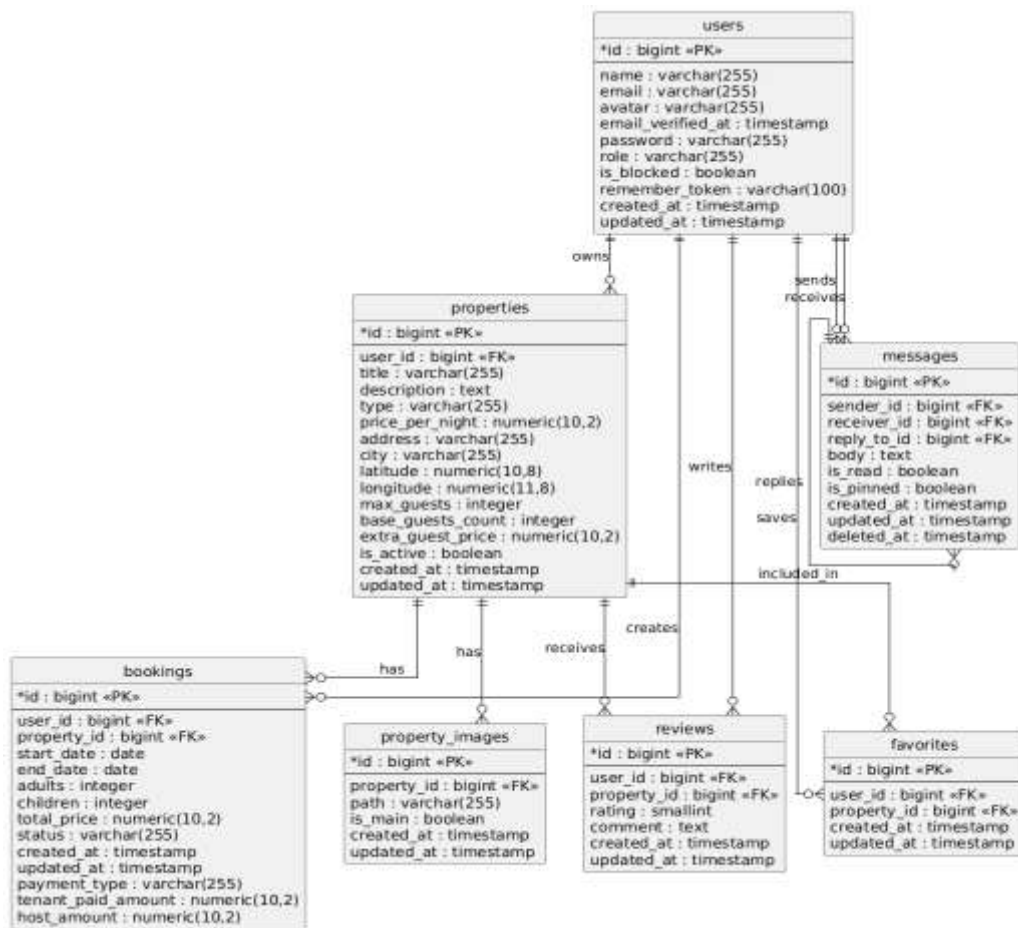


Рис. 2.3. ER-діаграма бази даних вебсервісу

ER-діаграма відображає структуру БД вебсервісу та зв'язки між основними сутностями системи. Центральне місце займають таблиці users, properties і bookings, які забезпечують роботу з користувачами, оголошеннями та бронюваннями. Детальний опис структури основних таблиць наведено далі.

Основною таблицею бази даних є users, яка зберігає інформацію про користувачів системи. У цій таблиці містяться ім'я користувача, електронна пошта, пароль, роль, ознака блокування та аватар. Поле role використовується для розмежування прав доступу між звичайним користувачем і адміністратором. Поле is_blocked дає змогу обмежувати доступ користувача до системи у разі порушення правил користування вебсервісом.

Таблиця `properties` призначена для зберігання інформації про оголошення житла. Вона містить зовнішній ключ `user_id`, який пов'язує оголошення з власником. До основних атрибутів таблиці належать назва, опис, тип житла, ціна за ніч, адреса, місто, координати, максимальна кількість гостей, статус активності, базова кількість гостей і доплата за додаткового гостя. Поле `is_active` використовується для контролю відображення оголошення у відкритій частині вебсервісу.

Для зберігання фотографій житла використовується таблиця `property_images`. Вона містить зовнішній ключ `property_id`, шлях до файлу зображення та ознаку головної фотографії. Винесення фотографій в окрему таблицю є доцільним, оскільки одне оголошення може мати декілька зображень. Це дозволяє формувати галерею фотографій і визначати основне зображення для відображення в картках житла.

Таблиця `bookings` використовується для зберігання бронювань. Вона містить зв'язки з користувачем і об'єктом житла через поля `user_id` та `property_id`. Також у таблиці зберігаються дати заїзду та виїзду, кількість дорослих і дітей, загальна вартість, статус бронювання, сума, зафіксована за орендарем, і сума, що нараховується власнику житла. Поле `status` використовується для визначення поточного стану бронювання, зокрема підтвердження або скасування.

Таблиця `reviews` призначена для зберігання відгуків користувачів про житло. Вона містить зовнішні ключі `user_id` і `property_id`, оцінку та текстовий коментар. Така структура дозволяє пов'язати кожен відгук із конкретним користувачем і конкретним об'єктом житла. Наявність відгуків підвищує інформативність оголошень і допомагає користувачам оцінювати якість житла перед бронюванням.

Для реалізації функції обраного використовується таблиця `favorites`. Вона зберігає зв'язок між користувачем і об'єктом житла, який було додано до списку обраного. Поля `user_id` і `property_id` дозволяють визначити, який саме користувач зберіг конкретне оголошення. Для запобігання дублюванню записів у таблиці доцільно передбачити унікальність пари `user_id` і `property_id`.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

Таблиця messages забезпечує роботу внутрішньої системи повідомлень. Вона містить ідентифікатор відправника, ідентифікатор отримувача, текст повідомлення, посилання на попереднє повідомлення у разі відповіді, статус прочитання, ознаку закріплення та дату м'якого видалення. Завдяки полям sender_id і receiver_id повідомлення пов'язується з користувачами системи, а поле is_read дає змогу відображати статус прочитання.

Запропонована структура БД забезпечує цілісне зберігання даних і підтримує основні процеси вебсервісу бронювання житла. Використання зовнішніх ключів дозволяє підтримувати зв'язки між сутностями та уникати дублювання інформації. Окреме зберігання користувачів, оголошень, бронювань, повідомлень, відгуків і обраних об'єктів спрощує подальше опрацювання даних у системі. Така структура дає змогу коректно виконувати пошук житла, створення бронювань, обмін повідомленнями, формування відгуків і розрахунок фінансових показників. Крім того, вона створює основу для подальшого розширення функціональності вебсервісу без суттєвої зміни основних зв'язків між таблицями.

Детальну структуру основних таблиць БД наведено в табл. 2.2–2.8. Першою є таблиця users, яка зберігає дані користувачів: ім'я, електронну пошту, пароль, роль у системі, ознаку блокування та шлях до аватара. Структуру таблиці users наведено в табл. 2.2.

Таблиця 2.2

Структура таблиці «Users»

Назва поля	Тип даних	РК	FK	Опис
id	bigint	+	-	Унікальний ідентифікатор користувача
name	varchar	-	-	Ім'я користувача
email	varchar	-	-	Електронна адреса користувача
avatar	varchar	-	-	Шлях до аватара користувача
email_verified_at	timestamp	-	-	Дата підтвердження електронної пошти
password	varchar	-	-	Хешований пароль користувача

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

role	varchar	-	-	Роль користувача в системі
is_blocked	boolean	-	-	Ознака блокування користувача
remember_token	varchar	-	-	Токен запам'ятовування користувача
created_at	timestamp	-	-	Дата й час створення запису
updated_at	timestamp	-	-	Дата й час оновлення запису

Інформація про об'єкти житла зберігається в таблиці properties. Вона пов'язана з таблицею users за допомогою зовнішнього ключа user_id, що дозволяє визначити власника кожного оголошення. Структуру таблиці properties наведено в табл. 2.3.

Таблиця 2.3

Структура таблиці «Properties»

Назва поля	Тип даних	PK	FK	Опис
id	bigint	+	-	Унікальний ідентифікатор об'єкта житла
user_id	bigint	-	+	Ідентифікатор власника житла
title	varchar	-	-	Назва оголошення
description	text	-	-	Опис об'єкта житла
type	varchar	-	-	Тип житла
price_per_night	numeric	-	-	Вартість проживання за одну ніч
address	varchar	-	-	Адреса об'єкта
city	varchar	-	-	Місто розташування житла
latitude	numeric	-	-	Географічна широта
longitude	numeric	-	-	Географічна довгота
max_guests	integer	-	-	Максимальна кількість людей
is_active	boolean	-	-	Ознака активності оголошення
base_guests_count	integer	-	-	Базова кількість гостей
extra_guest_price	numeric	-	-	Доплата за додаткового гостя
created_at	timestamp	-	-	Дата та час створення запису
updated_at	timestamp	-	-	Дата та час оновлення запису

Для зберігання даних про бронювання використовується таблиця bookings. Вона пов'язує користувача з конкретним об'єктом житла та містить дати проживання, кількість гостей, статус бронювання і фінансові параметри розподілу суми між платформою та власником житла. Наявність окремого статусу дає змогу контролювати життєвий цикл бронювання та зберігати історію скасованих і завершених операцій без фізичного видалення відповідних записів. Структуру таблиці bookings наведено в табл. 2.4.

Таблиця 2.4

Структура таблиці «Bookings»

Назва поля	Тип даних	РК	FK	Опис
id	bigint	+	-	Унікальний ідентифікатор бронювання
user_id	bigint	-	+	Ідентифікатор користувача
property_id	bigint	-	+	Ідентифікатор об'єкта житла
start_date	date	-	-	Дата заїзду
end_date	date	-	-	Дата виїзду
adults	integer	-	-	Кількість дорослих
children	integer	-	-	Кількість дітей
total_price	decimal	-	-	Загальна вартість бронювання
status	varchar	-	-	Статус бронювання
payment_type	varchar	-	-	Тип фінансового розрахунку
tenant_paid_amount	decimal	-	-	Сума, зафіксована за орендарем
host_amount	numeric	-	-	Сума, що нараховується власнику житла
created_at	timestamp	-	-	Дата й час створення запису
updated_at	timestamp	-	-	Дата й час оновлення запису

Фотографії об'єктів житла зберігаються в окремій таблиці property_images. Такий підхід дає змогу додавати до одного оголошення декілька зображень і визначати головне фото для відображення в каталозі. Кожний запис таблиці пов'язується з відповідним оголошенням за допомогою зовнішнього ключа. Структуру таблиці property_images наведено в табл. 2.5.

Таблиця 2.5

Структура таблиці «Property_images»

Назва поля	Тип даних	РК	FK	Опис
id	bigint	+	-	Унікальний ідентифікатор зображення
property_id	bigint	-	+	Ідентифікатор об'єкта житла
path	varchar	-	-	Шлях до файлу зображення
is_main	boolean	-	-	Ознака головного зображення
created_at	timestamp	-	-	Дата й час створення запису
updated_at	timestamp	-	-	Дата й час оновлення запису

Для зберігання відгуків користувачів про житло використовується таблиця reviews. Вона дозволяє пов'язати оцінку та коментар із конкретним користувачем і конкретним об'єктом житла. Структуру таблиці reviews наведено в табл. 2.6.

Таблиця 2.6

Структура таблиці «Reviews»

Назва поля	Тип даних	РК	FK	Опис
id	bigint	+	-	Унікальний ідентифікатор відгуку
user_id	bigint	-	+	Ідентифікатор користувача
property_id	bigint	-	+	Ідентифікатор об'єкта житла
rating	smallint	-	-	Оцінка житла
comment	text	-	-	Текст відгуку
created_at	timestamp	-	-	Дата й час створення запису
updated_at	timestamp	-	-	Дата й час оновлення запису

Для реалізації функції обраного використовується таблиця favorites. Вона зберігає інформацію про оголошення, які користувач додав до власного списку обраного. Структуру таблиці favorites наведено в табл. 2.7.

Таблиця 2.7

Структура таблиці «Favorites»

Назва поля	Тип даних	РК	FK	Опис
id	bigint	+	-	Унікальний ідентифікатор запису
user_id	bigint	-	+	Ідентифікатор користувача

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

property_id	bigint	-	+	Ідентифікатор об'єкта житла
created_at	timestamp	-	-	Дата й час створення запису
updated_at	timestamp	-	-	Дата й час оновлення запису

Для обміну повідомленнями між користувачами використовується таблиця messages. Вона забезпечує зберігання тексту повідомлень, інформації про відправника та отримувача, а також статусу прочитання. Структуру таблиці messages наведено в табл. 2.8.

Таблиця 2.8

Структура таблиці «Messages»

Назва поля	Тип даних	РК	ФК	Опис
id	bigint	+	-	Унікальний ідентифікатор повідомлення
sender_id	bigint	-	+	Ідентифікатор відправника
receiver_id	bigint	-	+	Ідентифікатор отримувача
reply_to_id	bigint	-	+	Ідентифікатор повідомлення, на яке надано відповідь
body	text	-	-	Текст повідомлення
is_read	boolean	-	-	Ознака прочитання повідомлення
is_pinned	boolean	-	-	Ознака закріплення повідомлення
deleted_at	timestamp	-	-	Дата м'якого видалення
created_at	timestamp	-	-	Дата й час створення запису
updated_at	timestamp	-	-	Дата й час оновлення запису

Зв'язки між таблицями побудовано відповідно до логіки предметної області. Таблиця users має зв'язок «один до багатьох» із таблицею properties, оскільки один користувач може створити декілька оголошень. Також користувач може мати багато бронювань, відгуків, обраних об'єктів і повідомлень.

Таблиця properties має зв'язок «один до багатьох» із таблицями property_images, bookings і reviews, оскільки одне оголошення може містити декілька фотографій, бронювань і відгуків. Зв'язок між користувачами й обраними оголошеннями реалізовано через таблицю favorites.

Для повідомлень використано два зовнішні ключі до таблиці users: sender_id і receiver_id. Поле reply_to_id створює самозв'язок таблиці messages, завдяки чому повідомлення може бути відповіддю на інше повідомлення.

Отже, розроблена структура БД забезпечує зберігання даних, необхідних для роботи вебсервісу: користувачів, оголошень, фотографій, бронювань, повідомлень, обраних об'єктів і відгуків.

2.3. Проєктування та реалізація алгоритмів роботи системи

Далі розглянемо логічну структуру та алгоритмічне забезпечення вебсервісу. Центральним елементом функціонування системи є забезпечення коректності процесу бронювання, що передбачає автоматичну перевірку доступності об'єкта нерухомості на обраний часовий проміжок.

Проєктування алгоритмів роботи вебсервісу є необхідним етапом, оскільки система повинна не лише зберігати дані, а й забезпечувати коректну послідовність виконання основних операцій. До ключових алгоритмів належать створення бронювання, перевірка перетину дат, автоматичний розрахунок вартості проживання, умовна оплата, скасування бронювання, створення та модерація оголошень, залишення відгуків і обмін повідомленнями між користувачами.

Найважливішим процесом у системі є створення бронювання. Він починається з того, що авторизований користувач відкриває сторінку житла, обирає дати заїзду та виїзду, зазначає кількість дорослих і дітей, після чого переходить до підтвердження бронювання. На цьому етапі система перевіряє коректність вхідних даних, доступність об'єкта на вибраний період і розраховує підсумкову вартість проживання.

Алгоритм створення бронювання вебсервісу наведено на рис. 2.4.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

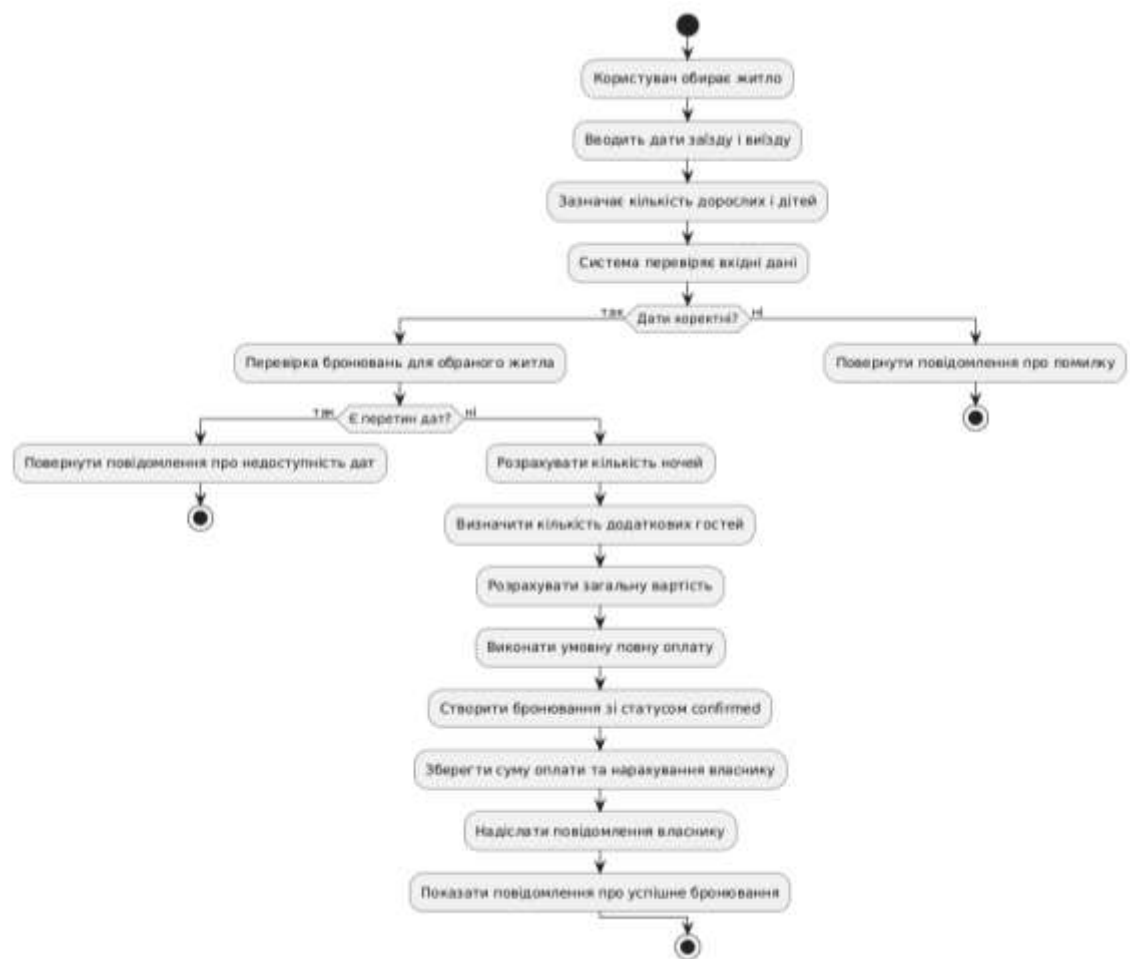


Рис. 2.4. Алгоритм створення бронювання у вебсервісі

Під час створення бронювання система спочатку перевіряє наявність об'єкта житла, коректність дат і кількість гостей. Дата виїзду повинна бути пізнішою за дату заїзду, а бронювання на минулі дати не допускається. Після 22:00 додатково забороняється створення бронювання на поточну дату. У такому випадку система пропонує обрати дату заїзду, починаючи з наступного дня. Бронювання на майбутні дати залишається доступним незалежно від часу оформлення.

Після перевірки дат виконується пошук перетину з уже створеними бронюваннями для того самого об'єкта. У системі враховуються всі бронювання, статус яких не є скасованим. Перетин визначається за умовою, коли дата початку нового бронювання є меншою за дату завершення існуючого бронювання, а дата завершення нового бронювання є більшою за дату початку існуючого бронювання. Такий підхід дозволяє заборонити накладання періодів і водночас

дозволити суміжні бронювання, коли дата виїзду одного користувача збігається з датою заїзду іншого.

Якщо перетин дат виявлено, система не створює бронювання та повертає користувачеві повідомлення про необхідність вибору іншого періоду. Якщо вибрані дати доступні, система визначає кількість ночей, кількість гостей, базову ціну за ніч, базову кількість гостей і доплату за додаткових гостей. Після цього обчислюється загальна сума бронювання.

Загальна вартість проживання визначається за формулою:

$$S = N \cdot (P + G_d \cdot P_d),$$

де S — загальна вартість бронювання; N — кількість ночей; P — базова ціна за ніч; G_d — кількість додаткових гостей; P_d — доплата за одного додаткового гостя за одну ніч.

Після розрахунку вартості система виконує умовну повну оплату без інтеграції реальної платіжної системи. У базі даних зберігається загальна сума, сума, сплачена користувачем, та сума, що нараховується власнику житла. Після створення бронювання власнику автоматично надсилається повідомлення із зазначенням орендаря, об'єкта, періоду проживання, кількості ночей, кількості гостей і загальної суми. Це забезпечує автоматизоване інформування власника про нове бронювання.

Окремим процесом є скасування бронювання. Скасувати бронювання може лише користувач, який його створив. Якщо бронювання вже має статус скасованого, повторне скасування не виконується. Також система не дозволяє скасовувати завершені бронювання, тобто ті, дата виїзду яких уже настала або минула. У разі успішного скасування статус бронювання змінюється на cancelled, а власнику житла надсилається повідомлення про те, що бронювання скасовано і відповідні дати знову доступні.

Алгоритм скасування бронювання наведено на рис. 2.5.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

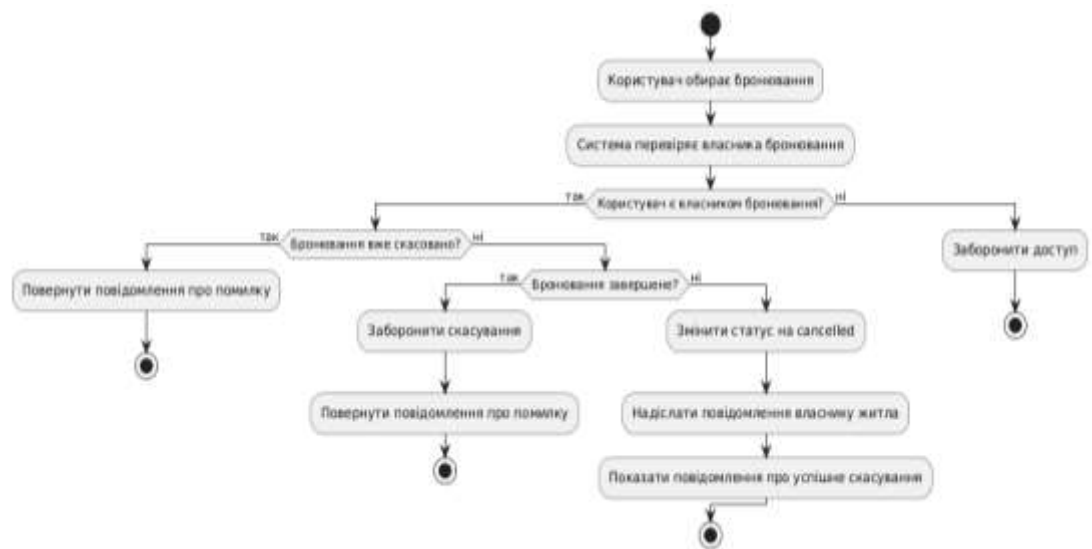


Рис. 2.5. Алгоритм скасування бронювання у вебсервісі

Важливою частиною роботи системи є створення та модерація оголошень. Авторизований користувач може створити оголошення про житло, вказавши назву, опис, місто, координати, ціну за ніч, базову кількість гостей, доплату за додаткового гостя та додавши фотографії. Під час створення оголошення система перевіряє введені дані, зокрема коректність назви, міста, числових значень і наявність щонайменше трьох фотографій. Після створення оголошення воно не активується автоматично, а отримує неактивний статус і надсилається на перевірку адміністратору.

Адміністратор переглядає оголошення та приймає рішення щодо його активації або деактивації. У разі активації оголошення стає доступним для перегляду та бронювання, а власнику надсилається повідомлення про успішну перевірку. У разі деактивації адміністратор повинен зазначити причину, після чого власник отримує повідомлення з поясненням. Такий алгоритм забезпечує контроль якості оголошень і дає змогу підтримувати актуальність даних у системі.

Використання модерації є важливим для системи, оскільки оголошення впливають на якість пошуку та довіру користувачів до вебсервісу. Перевірка адміністратором дає змогу виявити неповні описи, некоректні ціни, відсутність достатньої кількості фотографій або помилки в розташуванні житла. Завдяки

цьому у відкритій частині системи відображаються лише ті оголошення, які відповідають встановленим вимогам.

Алгоритм створення та модерації оголошення подано на рис. 2.6.

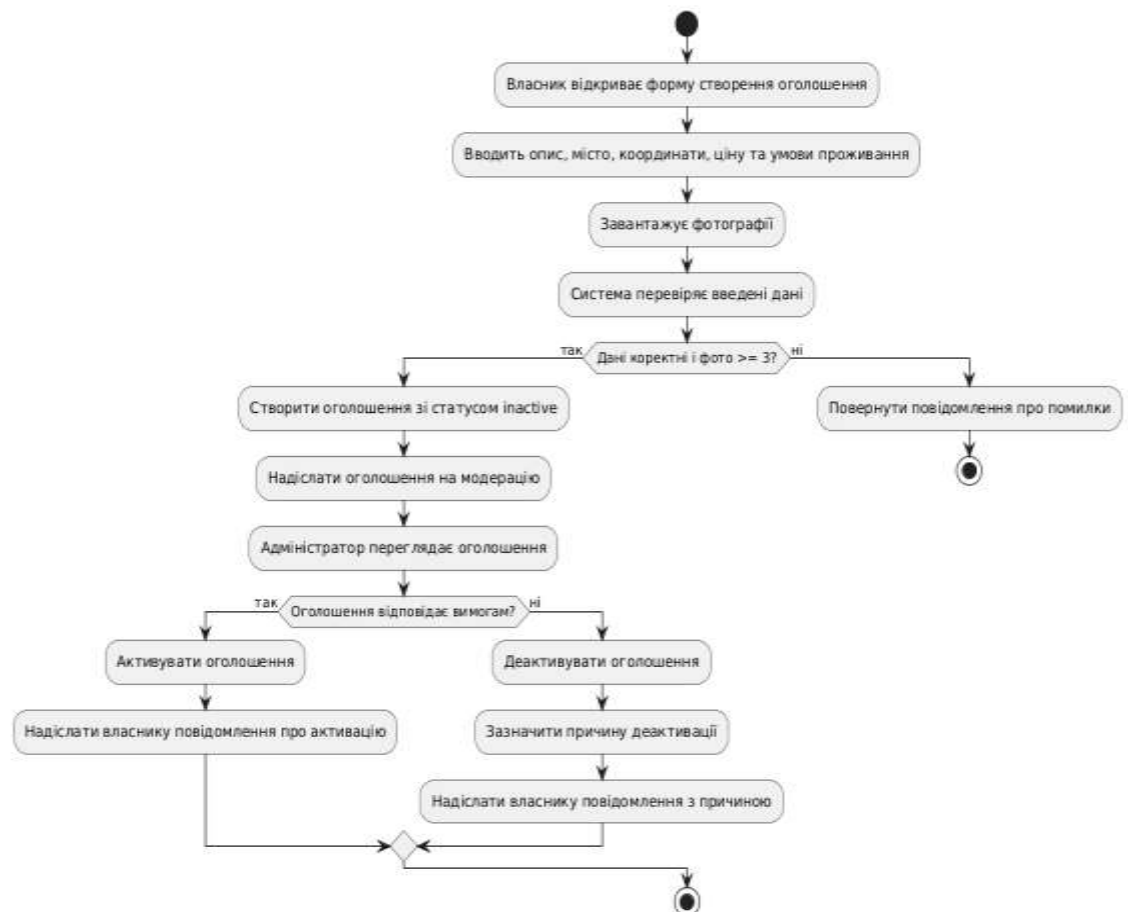


Рис. 2.6. Алгоритм створення та модерації оголошення у вебсервісі

Для підвищення достовірності відгуків у вебсервісі реалізовано обмеження на їх створення. Користувач може залишити відгук лише за умови, що він має підтверджене бронювання відповідного житла і дата завершення проживання вже минула. Якщо ці умови не виконані, система не дозволяє створити відгук і повертає повідомлення про помилку. Після успішної перевірки користувач вводить оцінку та текстовий коментар, які зберігаються в таблиці reviews і відображаються на сторінці житла.

Збереження відгуків у таблиці reviews дозволяє накопичувати оцінки для кожного об'єкта житла та використовувати їх під час формування рейтингу оголошення. Це підвищує інформативність сторінки житла й допомагає іншим користувачам приймати рішення перед бронюванням.

Алгоритм створення відгуку наведено на рис. 2.7.

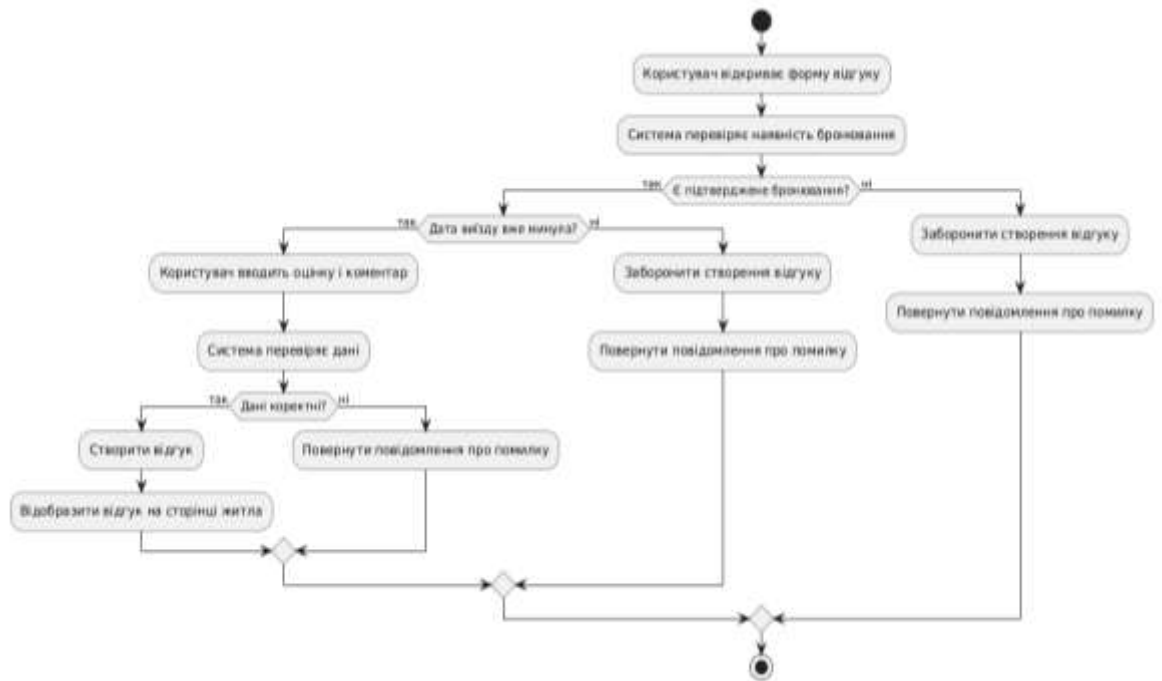


Рис. 2.7. Алгоритм створення відгуку після завершення проживання

Окремим алгоритмом є обмін повідомленнями. У системі повідомлення можуть використовуватися для спілкування між користувачами, власниками житла та адміністрацією. Під час надсилання повідомлення система перевіряє отримувача, текст повідомлення та, за наявності, посилання на повідомлення, на яке надається відповідь. Після збереження повідомлення воно отримує статус непрочитаного. Коли отримувач відкриває діалог, система позначає відповідні повідомлення як прочитані. Також у системі передбачено звернення користувача до підтримки, коли повідомлення автоматично спрямовується адміністратору. Реалізований механізм повідомлень використовується не лише для звичайного обміну інформацією між користувачами, а й для автоматичного інформування про важливі події в системі. Зокрема, власники житла отримують повідомлення про нові бронювання, їх скасування, результати модерації оголошень та інші зміни, що потребують уваги. Завдяки цьому користувачі можуть оперативно реагувати на події без необхідності постійного перегляду особистого кабінету.

Крім того, система підтримує відображення часу надсилання повідомлення, статусу його прочитання та кількості непрочитаних повідомлень.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

Це підвищує зручність комунікації та дозволяє користувачам швидко відстежувати нові звернення й відповіді в межах вебсервісу.

Алгоритм обміну повідомленнями у вебсервісі подано на рис. 2.8.

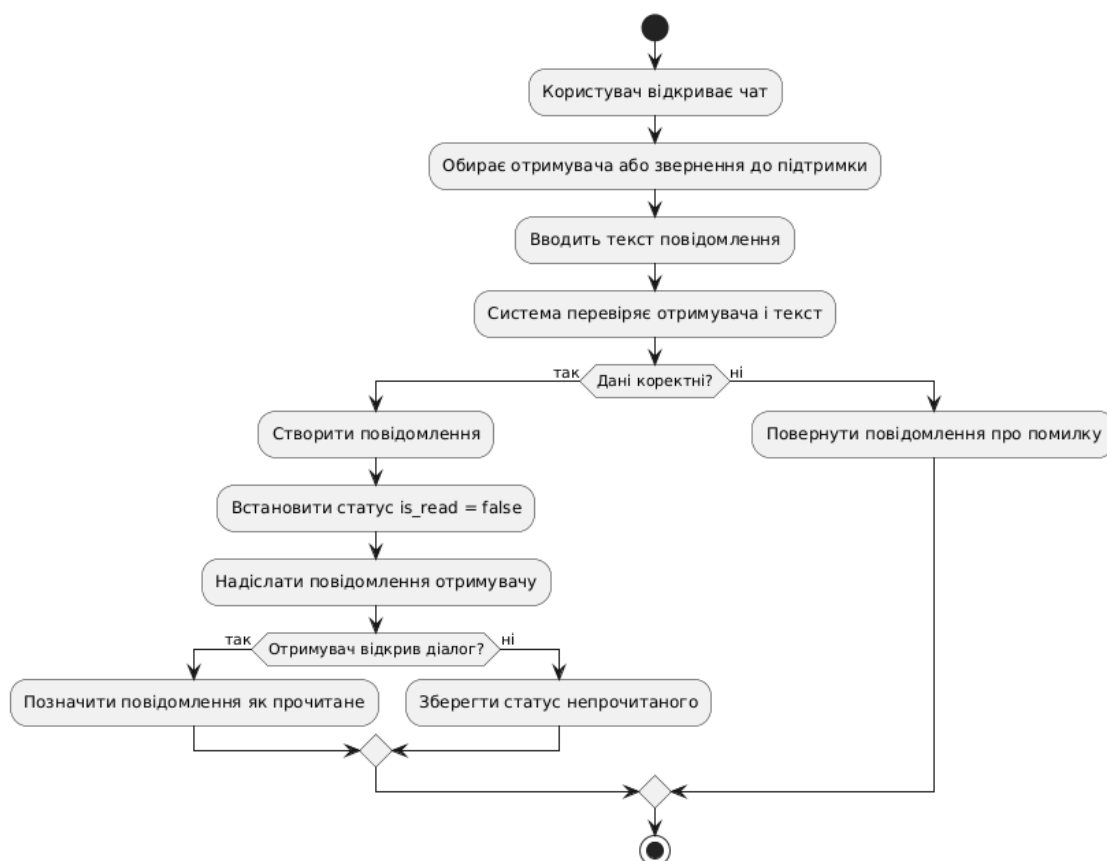


Рис. 2.8. Алгоритм обміну повідомленнями у вебсервісі

Розглянуті алгоритми забезпечують основну бізнес-логіку вебсервісу. Перевірка доступності дат запобігає накладанню бронювань, автоматичний розрахунок вартості зменшує ризик помилок під час оплати, модерація оголошень підтримує якість контенту, обмеження на створення відгуків підвищує достовірність оцінювання, а внутрішня система повідомлень забезпечує комунікацію між учасниками платформи. Таким чином, спроектовані алгоритми відповідають функціональним вимогам системи та створюють основу для реалізації її основних модулів.

2.4. Проектування бізнес-правил вебсервісу

Для коректної роботи вебсервісу недостатньо лише визначити структуру бази даних і основні алгоритми. Важливим етапом є формалізація бізнес-правил,

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

які визначають допустимі дії користувачів, обмеження системи та умови переходу об'єктів між станами. Такі правила забезпечують цілісність даних, запобігають конфліктам бронювання та визначають порядок взаємодії між орендарем, власником житла й адміністратором [19].

Першим ключовим правилом є перевірка доступності житла на вибраний період. Нове бронювання може бути створене лише за умови, що для відповідного об'єкта відсутні активні бронювання з перетином дат. При цьому скасовані бронювання не враховуються під час перевірки доступності. Суміжні бронювання дозволяються, тобто дата виїзду одного користувача може збігатися з датою заїзду іншого.

Другим правилом є обмеження щодо скасування бронювання. Користувач може скасувати лише власне бронювання, якщо воно ще не завершене. Завершені бронювання не скасовуються, оскільки період проживання вже минув і запис має залишатися в історії. Скасовані бронювання не видаляються з бази даних, а лише змінюють статус, що дає змогу зберігати історію взаємодії користувача із системою.

Третє правило стосується розрахунку вартості. Підсумкова сума визначається автоматично на серверному рівні з урахуванням кількості ночей, базової ціни за ніч, базової кількості гостей і доплати за додаткових гостей. Такий підхід зменшує ризик помилок і не дозволяє користувачеві змінити суму лише на рівні клієнтського інтерфейсу.

Окреме бізнес-правило стосується створення оголошень. Власник житла повинен заповнити основні характеристики об'єкта та додати не менше трьох фотографій. Після створення оголошення не активується автоматично, а передається на перевірку адміністратору. Адміністратор може активувати оголошення або деактивувати його із зазначенням причини.

Для відгуків передбачено обмеження, відповідно до якого користувач може залишити оцінку лише після завершення проживання. Це правило підвищує достовірність відгуків, оскільки оцінювання виконується користувачем, який фактично мав завершене бронювання відповідного житла.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

Також у системі передбачено автоматичне створення повідомлень про важливі події. Власник отримує повідомлення про нове бронювання або його скасування, а користувач може звернутися до адміністратора через внутрішню систему підтримки. Повідомлення мають статус прочитання, що дозволяє відстежувати нові звернення.

Окрему групу становлять правила, пов'язані з правами доступу користувачів. Неавторизований користувач може переглядати відкриту частину вебсервісу, виконувати пошук житла та відкривати сторінки оголошень. Однак створення бронювання, додавання житла до обраного, надсилання повідомлень, створення оголошень і залишення відгуків доступні лише після автентифікації. Такий підхід забезпечує контроль виконання персоналізованих дій і дозволяє пов'язувати кожну операцію з конкретним користувачем.

Для адміністратора передбачено окремі правила доступу до функцій керування системою. Адміністратор може переглядати користувачів, бронювання, оголошення, повідомлення та статистичні показники, а також змінювати статуси оголошень і бронювань. Звичайні користувачі не повинні мати доступу до адміністративної панелі, оскільки це може призвести до несанкціонованої зміни даних. Тому всі адміністративні дії мають виконуватися лише після перевірки ролі користувача [29].

Важливими є також правила, що стосуються фінансових параметрів бронювання. Після підтвердження бронювання система фіксує загальну суму оплати користувачем і суму, що нараховується власнику житла. У межах реалізованого вебсервісу власнику нараховується 90 % загальної вартості бронювання, а решта 10 % становить комісію платформи. Значення комісії визначається як різниця між загальною сумою оплати та сумою, нарахованою власнику. Збережені фінансові параметри використовуються для відображення показників в особистому кабінеті власника та адміністративній панелі.

Ще одним правилом є збереження історії взаємодії користувачів із системою. Бронювання, повідомлення, відгуки та оголошення не повинні втрачати зв'язок із користувачами й об'єктами житла, оскільки ці дані

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

використовуються для перегляду історії поїздок, формування рейтингу, аналізу активності та підготовки статистичних показників. Це забезпечує послідовність роботи вебсервісу та дає змогу надалі розширювати аналітичні можливості системи.

Основні бізнес-правила вебсервісу наведено в табл. 2.9.

Таблиця 2.9

Основні бізнес-правила вебсервісу

№	Бізнес-правило	Умови виконання	Результат
1.	Створення бронювання	Дати коректні та не перетинаються з активними бронюваннями	Бронювання створюється
2.	Заборона перетину дат	Обраний період перетинається з активним бронюванням	Система блокує бронювання
3.	Суміжні бронювання	Дата виїзду одного бронювання збігається з датою заїзду іншого	Бронювання дозволяється
4.	Скасування бронювання	Бронювання належить користувачу та ще не завершено	Статус змінюється на cancelled
5.	Завершене бронювання	Дата виїзду вже настала або минула	Скасування забороняється
6.	Створення оголошення	Заповнено основні поля та додано не менше трьох фотографій	Оголошення надсилається на модерацію
7.	Модерація оголошення	Адміністратор перевірів дані	Оголошення активується або деактивується
8.	Створення відгуку	Період проживання завершено	Відгук зберігається та відображається
9.	Повідомлення власнику	Створено або скасовано бронювання	Власнику надсилається системне повідомлення
10.	Фінансовий розрахунок	Бронювання підтверджено	Зберігається загальна сума та нарахування власнику

Отже, визначені бізнес-правила формують логічну основу роботи вебсервісу. Вони регламентують створення бронювань, перевірку доступності дат, розрахунок вартості, скасування бронювань, модерацію оголошень, залишення відгуків і роботу повідомлень. Формалізація цих правил підвищує надійність системи та забезпечує відповідність реалізованого функціоналу поставленим вимогам.

2.5. Реалізація функціоналу вебсервісу

Функціонал вебсервісу реалізовано на серверному та клієнтському рівнях. Користувацький інтерфейс побудовано з Vue-компонентів, а основні операції опрацьовуються контролерами Laravel. Взаємодія між частинами системи здійснюється за допомогою Inertia.js. Маршрути бронювання, повідомлень, відгуків, обраного та керування оголошеннями доступні лише автентифікованим користувачам.

Основним функціональним процесом є створення бронювання. Запит надходить до методу store контролера BookingController, який перевіряє вхідні дані та доступність житла, розраховує вартість і створює запис у базі даних.

Спочатку виконується валідація ідентифікатора житла, дат заїзду й виїзду, кількості дорослих і дітей та типу фінансового розрахунку [31]. Ідентифікатор повинен відповідати наявному об'єкту, кількість дорослих не може бути меншою за одиницю, а кількість дітей — від'ємною. Загальна вартість визначається сервером, тому значення, отримане від клієнтської частини, не використовується як остаточне. Фрагмент валідації наведено в додатку А, лістинг А.1.

Далі перевіряється коректність дат. Дата виїзду повинна бути пізнішою за дату заїзду, а бронювання на минулий період не допускається. Після 22:00 система додатково забороняє створення бронювання на поточну дату, водночас бронювання на наступні дні залишається доступним. Фрагмент перевірки дат наведено в додатку А, лістинг А.2.

Після цього система визначає, чи немає активного бронювання на той самий період. Ураховуються записи, статус яких відрізняється від cancelled.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

Перетин існує, якщо дата початку наявного бронювання є меншою за нову дату виїзду, а дата завершення — більшою за нову дату заїзду. Строгі оператори порівняння дозволяють створювати суміжні бронювання, коли дата виїзду одного користувача збігається з датою заїзду наступного. Фрагмент перевірки перетину дат наведено в додатку А, лістинг А.3.

Якщо на вибраний період уже існує активне бронювання, створення нового запису припиняється, а користувач отримує повідомлення про необхідність вибору інших дат. Введені значення форми зберігаються.

Після перевірки доступності система визначає кількість ночей, базову вартість і доплату за додаткових гостей. Загальна кількість гостей обчислюється як сума дорослих і дітей. Базова вартість дорівнює добутку ціни за ніч і кількості ночей. Якщо кількість гостей перевищує встановлене базове значення, розраховується додаткова плата за весь період проживання. Фрагмент розрахунку вартості наведено в додатку А, лістинг А.4.

Після визначення загальної суми виконується умовний фінансовий розподіл: повна сума фіксується за орендарем, 90 % нараховується власнику житла, а 10 % становить комісію платформи. Реальна платіжна система в межах роботи не підключається.

Після завершення перевірок у таблиці bookings створюється запис, що містить ідентифікатори користувача та житла, дати проживання, кількість гостей, загальну суму, тип фінансового розрахунку, суми орендаря й власника та статус confirmed. Фрагмент створення бронювання наведено в додатку А, лістинг А.5.

Після успішного бронювання система формує внутрішнє повідомлення власнику житла з даними про орендаря, об'єкт, період проживання, кількість гостей і загальну суму. Якщо користувач бронює власний об'єкт, повідомлення не створюється. Запис зберігається в таблиці messages, після чого формується подія MessageSent, яка передається через приватний канал отримувача. У разі помилки трансляції інформація записується до журналу, але бронювання

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

зберігається. Фрагмент створення повідомлення наведено в додатку А, лістинг А.6.

Послідовність операцій під час створення бронювання наведено на рис. 2.9.

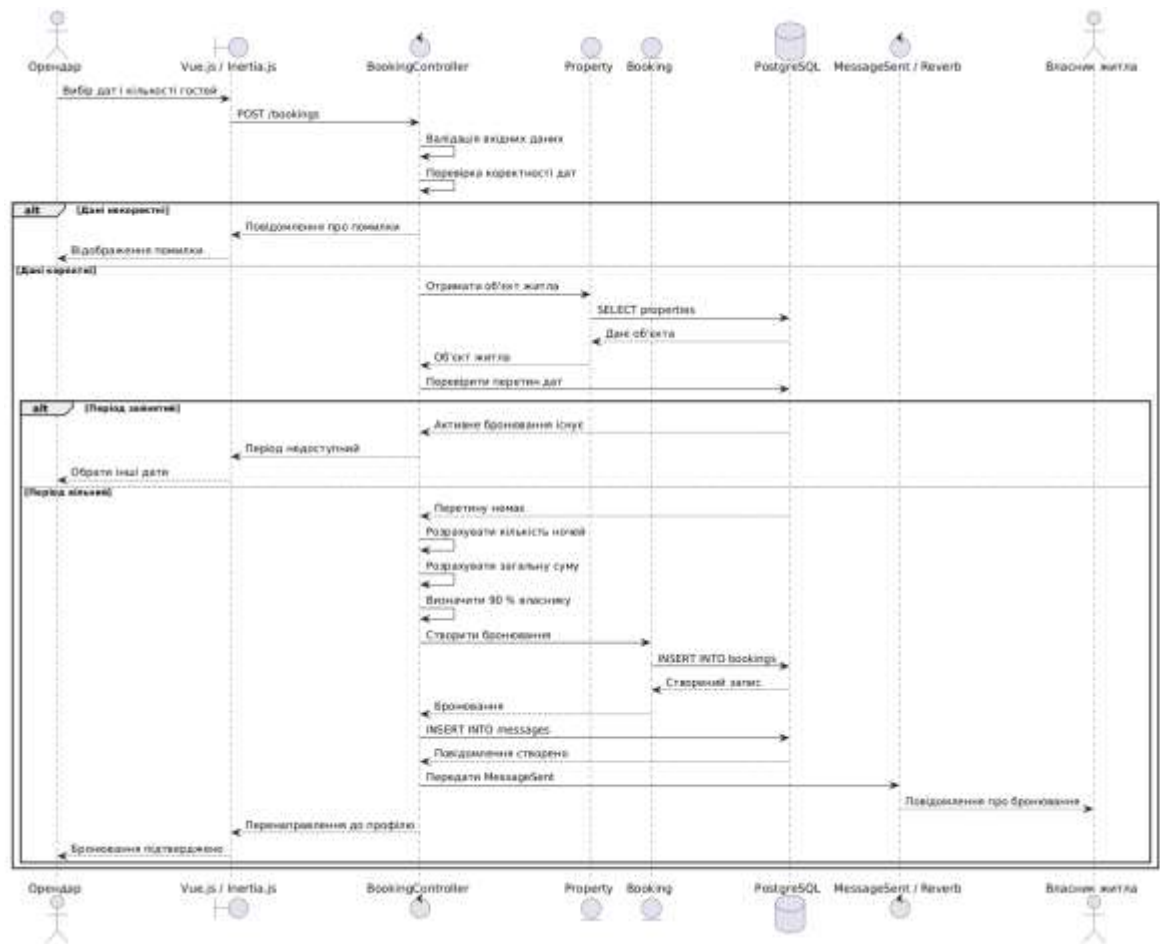


Рис. 2.9. Діаграма послідовності створення бронювання

Окремо реалізовано скасування бронювання. Перед зміною статусу система перевіряє, чи належить запис поточному користувачеві. Не допускається повторне скасування бронювання зі статусом cancelled, а також скасування після настання дати виїзду.

Після проходження перевірок статус змінюється на cancelled, але запис фізично не видаляється, що забезпечує збереження історії операцій. Власнику надсилається внутрішнє повідомлення, а звільнений період знову стає доступним для інших користувачів. Фрагмент скасування бронювання наведено в додатку А, лістинг А.7.

У вебсервісі також реалізовано створення відгуків. Залишити відгук може лише користувач, який має підтверджене бронювання відповідного житла із завершеним періодом проживання. Під час збереження перевіряються текст коментаря та оцінка від одного до п'яти. Фрагмент реалізації наведено в додатку А, лістинг А.8.

Додавання житла до обраного та видалення зі списку виконується методом toggle. Якщо зв'язок між користувачем і об'єктом уже існує, відповідний запис видаляється з проміжної таблиці, а за його відсутності — створюється. Фрагмент реалізації наведено в додатку А, лістинг А.9.

Для обміну інформацією реалізовано внутрішню систему повідомлень. Під час надсилання перевіряються отримувач, текст та, за наявності, повідомлення, на яке надається відповідь. Новий запис створюється з ознакою непрочитаного, після чого транслюється подія MessageSent. Також передбачено позначення повідомлень як прочитаних, редагування й видалення власних записів, їх закріплення та очищення історії діалогу. Фрагмент надсилання повідомлення наведено в додатку А, лістинг А.10.

Під час створення оголошення користувач зазначає назву, опис, тип житла, місто, адресу, ціну, кількість гостей, координати та параметри доплати. Фотографії зберігаються у файловому сховищі вебсервісу, а їхні шляхи — у таблиці property_images. Після створення оголошення отримує неактивний статус і не відображається у відкритому переліку до завершення модерації. Фрагмент створення оголошення наведено в додатку А, лістинг А.11.

Під час редагування система перевіряє належність оголошення поточному користувачеві. Підсумкова кількість фотографій визначається з урахуванням наявних, видалених і нових зображень. Якщо їх залишається менше трьох, зміни не зберігаються. Видалені файли та відповідні записи вилучаються, а відомості про нові фотографії додаються до таблиці property_images. Фрагмент перевірки кількості фотографій наведено в додатку А, лістинг А.12.

Модерацію оголошень виконує адміністратор. Після активації об'єкт стає доступним для пошуку й бронювання, а для власника створюється внутрішнє

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

повідомлення та формується поштове сповіщення. Під час деактивації адміністратор повинен зазначити причину, яка передається власникові через внутрішній чат. Фрагмент реалізації модерації наведено в додатку А, лістинг А.13.

Доступ до функцій вебсервісу регулюється маршрутами та проміжним програмним забезпеченням: операції з бронюваннями, відгуками, обраним, повідомленнями й оголошеннями доступні автентифікованим користувачам, а адміністративні маршрути додатково перевіряють роль admin. Для заблокованого користувача обмежуються основні операції, але зберігається можливість вийти з облікового запису та звернутися до адміністратора через чат. Фрагмент перевірки доступу наведено в додатку А, лістинг А.14.

Результати серверного опрацювання передаються до клієнтського інтерфейсу у вигляді HTTP-відповідей. Залежно від результату запиту сервер повертає код стану, який указує на успішне виконання операції, помилку вхідних даних або відсутність необхідних прав доступу [30]. У разі помилки операція не виконується, а після успішного створення, редагування або скасування запису система оновлює дані та повідомляє користувача про результат.

Отже, реалізований функціонал охоплює створення та модерацію оголошень, роботу з фотографіями, перевірку доступності житла, розрахунок вартості, створення і скасування бронювань, обмін повідомленнями, обране та відгуки з урахуванням установлених бізнес-правил і прав доступу.

Висновки до розділу 2

У другому розділі виконано проектування вебсервісу для бронювання житла та реалізовано його основні функціональні компоненти. Визначено три основні категорії користувачів системи: неавторизований користувач, авторизований користувач та адміністратор. Установлено, що авторизований користувач залежно від поточного сценарію може виконувати функції орендаря або власника житла. Для кожної категорії визначено доступні варіанти використання та відповідні обмеження.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

Спроектовано об'єктно-орієнтовану структуру вебсервісу на основі моделей User, Property, Booking, PropertyImage, Message, Review і Favorite. Визначено призначення основних класів та зв'язки між ними. Розроблено структуру бази даних для зберігання користувачів, оголошень, фотографій житла, бронювань, повідомлень, відгуків та обраних об'єктів.

Розглянуто алгоритми створення і скасування бронювання, перевірки доступності вибраного періоду, розрахунку вартості проживання та умовного розподілу коштів. Сформовано бізнес-правила, які визначають порядок роботи з датами, оголошеннями, відгуками, повідомленнями та правами доступу користувачів.

Реалізовано створення та редагування оголошень, збереження фотографій, модерацію об'єктів адміністратором, пошук і бронювання житла, скасування бронювань, додавання об'єктів до обраного, створення відгуків та обмін внутрішніми повідомленнями. Передбачено формування сповіщень про створення і скасування бронювання, активацію та деактивацію оголошень. Також реалізовано перевірку ролей користувачів і обмеження доступу для заблокованих облікових записів.

Отже, у межах розділу сформовано структуру вебсервісу, визначено правила його роботи та реалізовано основні програмні операції. Отримані результати створюють основу для подальшого опису користувацького інтерфейсу, порядку роботи із системою та проведення її тестування.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		60

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБСЕРВІСОМ

3.1. Порядок встановлення та налаштування параметрів системи

Реалізація вебсервісу виконана з використанням Laravel, Vue.js 3, Inertia.js, Tailwind CSS, PostgreSQL, Vite, Docker і Laravel Sail. Для коректної роботи системи необхідно підготувати середовище розробки, встановити залежності серверної та клієнтської частин, налаштувати підключення до бази даних і виконати запуск контейнеризованого середовища.

Проект має структуру Laravel-застосунку, у якому серверна логіка розміщується в каталогах `app`, `routes`, `database` та `config`, а клієнтська частина на основі Vue.js 3 та Inertia.js — у каталозі `resources/js`. Така структура забезпечує логічне розділення серверної частини, маршрутизації, міграцій бази даних і компонентів користувацького інтерфейсу.

Для запуску вебсервісу використовується Laravel Sail, який працює на основі Docker. Контейнеризація забезпечує роботу застосунку у стандартизованому середовищі незалежно від особливостей операційної системи розробника. Локальне середовище містить контейнер Laravel-застосунку, у якому запускається серверна частина та засоби збирання клієнтських ресурсів Vite, а також окремі сервіси PostgreSQL, Redis і Laravel Reverb. PostgreSQL використовується для зберігання даних, Redis входить до складу допоміжної інфраструктури, а Laravel Reverb забезпечує передавання подій у реальному часі.

Взаємодія між контейнерами відбувається через внутрішню мережу `'sail'`. Контейнер Laravel-застосунку звертається до PostgreSQL і Redis за назвами відповідних сервісів, тому для обміну даними між ними не потрібно використовувати зовнішні мережеві адреси. Сервіс Laravel Reverb запускається окремо та взаємодіє із застосунком для передавання подій, пов'язаних із повідомленнями в реальному часі. Така організація дає змогу незалежно запускати, зупиняти й перевіряти роботу кожного компонента середовища.

Схему локального розгортання вебсервісу наведено на рис. 3.1.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

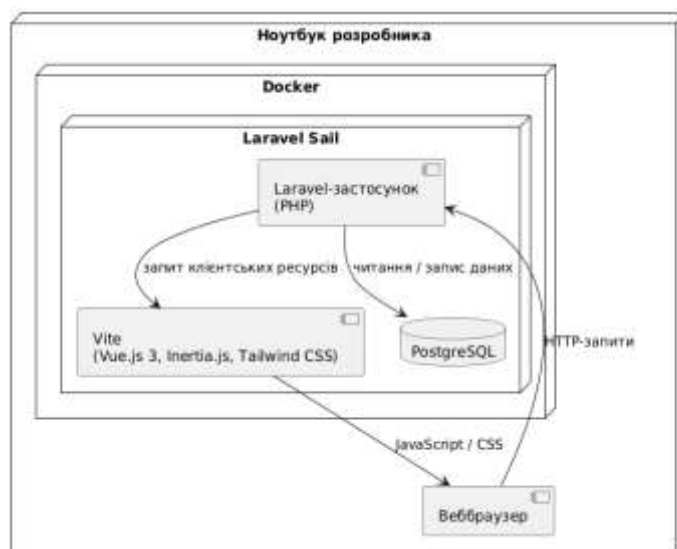


Рис. 3.1. Схема локального розгортання вебсервісу

Після підготовки локального середовища виконується встановлення та налаштування вебсервісу. На комп'ютері розробника повинні бути встановлені Docker, Git і Composer. Docker використовується для запуску контейнерів, Git — для отримання вихідного коду, а Composer — для встановлення залежностей серверної частини [15, 24].

У кореневому каталозі проєкту виконується команда `composer install`, яка опрацьовує файл `composer.json` і створює каталог `vendor`. Після цього на основі шаблону `.env.example` створюється файл `.env`, у якому задаються адреса застосунку, параметри бази даних і налаштування передавання подій. Командою `php artisan key:generate` формується ключ застосунку, необхідний для роботи механізмів Laravel [10].

Використання файлу `.env` дає змогу відокремити параметри середовища від програмного коду. У ньому зберігаються налаштування підключення до PostgreSQL, Redis і Laravel Reverb, а також адреси та порти сервісів. Це спрощує перенесення вебсервісу на інший комп'ютер без зміни основних файлів проєкту. Конфіденційні значення файлу `.env` не повинні додаватися до системи контролю версій.

Оскільки для зберігання даних використовується PostgreSQL, у файлі `.env` указується драйвер `pgsql`, назва сервісу `pgsql`, порт `5432`, назва бази даних, ім'я користувача та пароль. Значення параметрів повинні відповідати налаштуванням

контейнера PostgreSQL у файлі `compose.yaml`. PostgreSQL забезпечує зберігання даних та підтримку зв'язків між таблицями [13, 14].

Для роботи повідомлень у реальному часі налаштовується `Laravel Reverb`. У файлі `.env` задаються ідентифікатор застосунку, ключ, секретне значення, адреса та порт сервісу. Клієнтські параметри з префіксом `VITE_REVERB_` використовуються `Laravel Echo` для встановлення з'єднання з веббраузером.

Після завершення налаштування запускається контейнеризоване середовище командою `./vendor/bin/sail up -d`. `Laravel Sail` забезпечує керування Docker-контейнерами застосунку, PostgreSQL, Redis і `Laravel Reverb` [15, 16]. Стан запущених сервісів можна перевірити командою `./vendor/bin/sail ps`.

Далі командою `./vendor/bin/sail artisan migrate` створюється структура бази даних. Для забезпечення доступу до завантажених фотографій виконується команда `./vendor/bin/sail artisan storage:link`, яка створює посилання між файловим сховищем `Laravel` і каталогом `public/storage` [10].

Клієнтські залежності встановлюються командою `./vendor/bin/sail npm install`. `npm` працює в середовищі `Node.js` і опрацьовує перелік пакетів із файла `package.json` [26, 27]. Після встановлення залежностей виконується команда `./vendor/bin/sail npm run dev`, яка запускає `Vite` для оброблення `Vue`-компонентів, `JavaScript`-файлів і стилів [17].

Після виконання зазначених дій вебсервіс стає доступним у браузері за адресою `http://localhost:8080`. Під час перевірки необхідно переконатися, що сторінки завантажуються без помилок, підключення до PostgreSQL встановлено, фотографії відображаються, а повідомлення надходять у реальному часі.

Після першого запуску доцільно перевірити журнали роботи контейнерів за допомогою команди `./vendor/bin/sail logs`. Це дає змогу виявити помилки підключення до бази даних, неправильні параметри середовища або конфлікти портів. Для припинення роботи вебсервісу використовується команда `./vendor/bin/sail down`, яка зупиняє контейнери без видалення збережених даних. Використання `Docker` і `Laravel Sail` спрощує повторне розгортання вебсервісу та забезпечує однакові умови його роботи на різних комп'ютерах [15, 16].

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		63

3.2. Структура інтерфейсу та порядок взаємодії з вебсервісом

Користувацьку частину вебсервісу реалізовано на основі клієнт-серверної архітектури. Серверну частину розроблено мовою PHP із використанням Laravel, який забезпечує обробку HTTP-запитів, автентифікацію, роботу з PostgreSQL і виконання бізнес-логіки. Клієнтський інтерфейс побудовано за допомогою Vue.js 3, Inertia.js і Tailwind CSS, що дозволяє поєднати Laravel із динамічними Vue-компонентами без створення окремого REST API для кожної сторінки.

Неавторизований користувач може переглядати оголошення, користуватися пошуком і фільтрацією. Для бронювання, додавання житла до обраного, створення оголошень, надсилання повідомлень і залишення відгуків необхідно пройти автентифікацію.

Головна сторінка містить каталог житла, пошук за локацією, фільтрацію за бюджетом, сортування за новизною, вартістю та рейтингом, а також перехід до обраного. У картці житла відображаються фотографія, місто, назва, вартість, рейтинг і кнопка додавання до обраного. Загальний вигляд головної сторінки наведено на рис. 3.2.

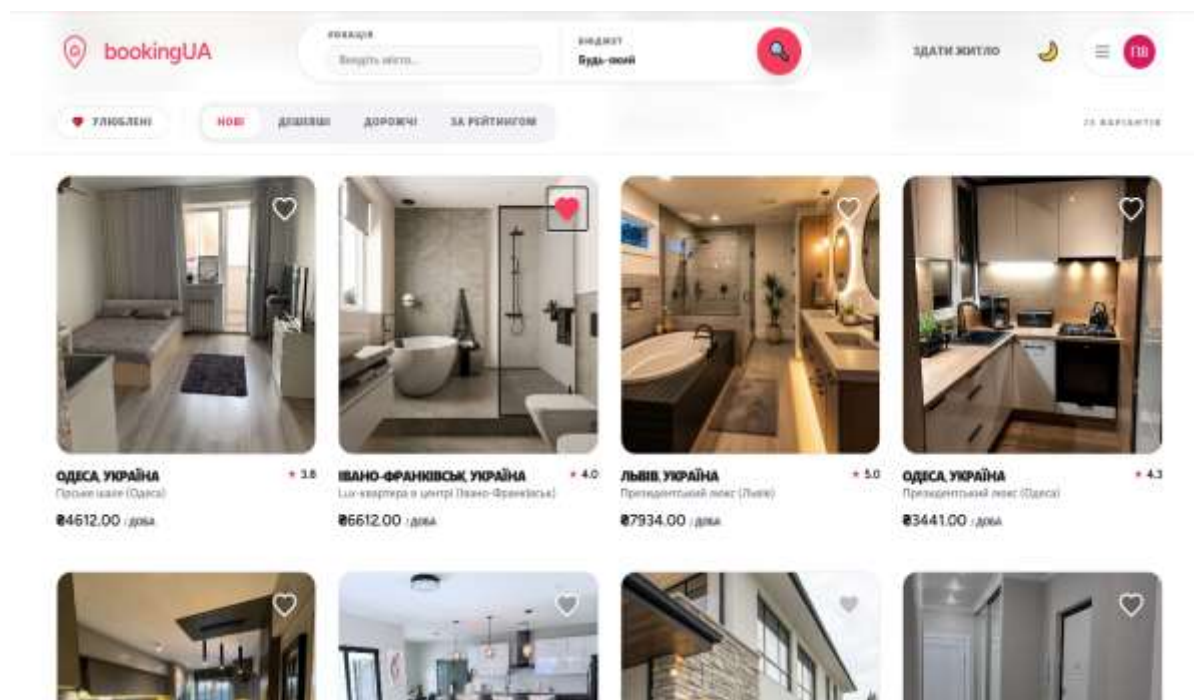


Рис. 3.2. Головна сторінка вебсервісу

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

Для підвищення зручності користування у вебсервісі реалізовано підтримку світлої та темної теми інтерфейсу. Перемикання теми виконується з верхньої панелі навігації, після чого змінюється оформлення основних сторінок, карток житла, елементів навігації та фільтрів. На головній сторінці також реалізовано пошук житла за містом та фільтрацію за максимальною вартістю проживання. Результати пошуку оновлюються відповідно до заданих параметрів, що спрощує вибір необхідного варіанта житла. Для кожного оголошення відображаються фотографія, назва, рейтинг, місце розташування та вартість проживання за добу. Крім того, авторизовані користувачі можуть додавати оголошення до списку обраного без переходу на сторінку перегляду об'єкта. Для впорядкування результатів користувач може сортувати оголошення за датою додавання, вартістю та середнім рейтингом, а також відображати лише об'єкти зі списку обраного. Після застосування параметрів інтерфейс показує кількість знайдених варіантів житла. За потреби встановлені фільтри можна скинути та повернутися до початкового переліку. Якщо жоден об'єкт не відповідає заданим умовам, виводиться відповідне інформаційне повідомлення. Головну сторінку вебсервісу у темній темі подано на рис. 3.3.

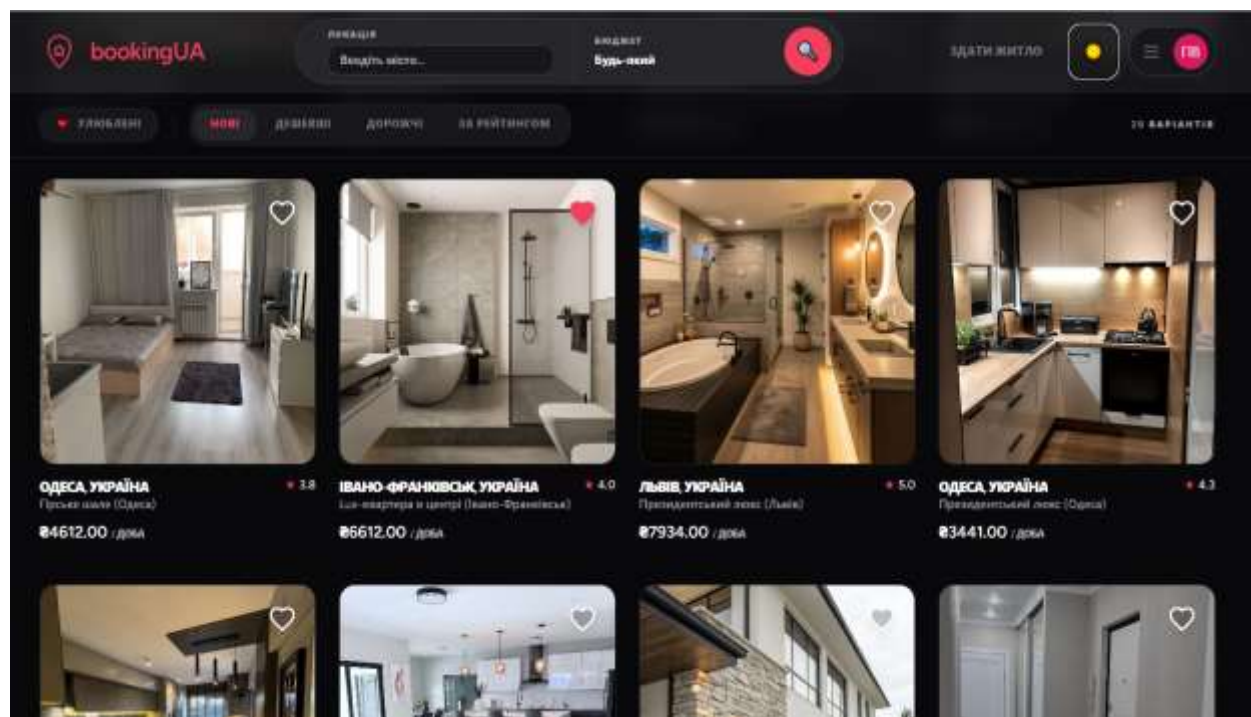


Рис. 3.3. Головна сторінка вебсервісу у темній темі

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

Після вибору житла користувач переходить на сторінку детального перегляду оголошення. На цій сторінці відображаються назва об'єкта, рейтинг, місто, галерея фотографій, інформація про власника, опис житла, форма бронювання, блок відгуків і карта розташування. Така структура сторінки дає змогу користувачу отримати всю необхідну інформацію перед оформленням бронювання. Використання темної теми дозволяє зменшити навантаження на зір користувача під час тривалої роботи із вебсервісом. Під час перемикання теми всі елементи інтерфейсу автоматично адаптуються до обраного режиму відображення. При цьому зберігається єдиний стиль оформлення сторінок та забезпечується належна читабельність текстової інформації. Обраний режим інтерфейсу зберігається для подальших відвідувань вебсервісу. На сторінці перегляду користувач також може ознайомитися з відгуками інших мешканців та середнім рейтингом об'єкта. Це дозволяє отримати додаткову інформацію про якість житла та умови проживання ще до оформлення бронювання. Для зручності користувачів на сторінці відображається карта із зазначенням місця розташування об'єкта, що допомагає оцінити його віддаленість від необхідної інфраструктури. Сторінку перегляду оголошення наведено на рис. 3.4.

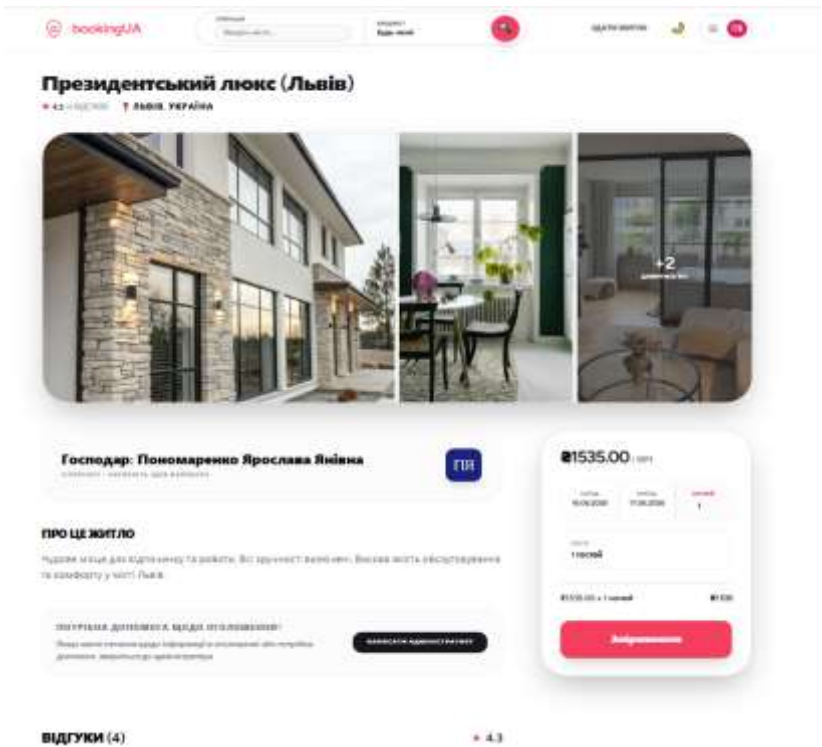


Рис. 3.4. Сторінка перегляду оголошення

Також на сторінці оголошення передбачено блок звернення до адміністратора, показаний на рис. 3.5. Він використовується у випадках, коли користувач має запитання щодо інформації в оголошенні або потребує додаткової допомоги. Після натискання кнопки «Написати адміністратору» користувач може перейти до внутрішнього чату та звернутися до адміністрації платформи. Це забезпечує додатковий канал підтримки користувачів безпосередньо зі сторінки житла.

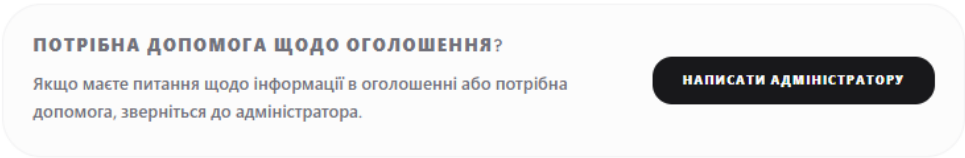


Рис. 3.5. Блок звернення до адміністратора на сторінці оголошення

У правій частині сторінки оголошення розміщено блок бронювання. Користувач обирає дату заїзду, дату виїзду та кількість гостей, після чого система автоматично визначає кількість ночей і загальну суму. Якщо кількість гостей перевищує базову кількість, до вартості додається доплата за додаткових гостей. Приклад автоматичного розрахунку вартості бронювання з урахуванням додаткового гостя наведено на рис. 3.6.

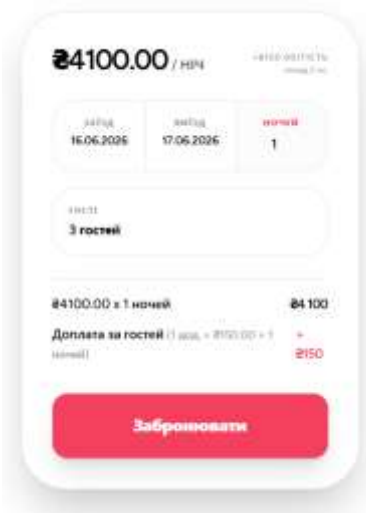


Рис. 3.6. Розрахунок вартості бронювання з урахуванням додаткових гостей

Після натискання кнопки бронювання користувач переходить на сторінку підтвердження замовлення. На ній відображаються короткі відомості про житло, місто, період проживання, кількість гостей, базова сума, доплата за додаткових

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

гостей і підсумкова вартість. У межах проєкту реалізовано умовну повну оплату через платформу без інтеграції реальної платіжної системи. Автоматичний розрахунок вартості виконується безпосередньо під час зміни параметрів бронювання. Це дозволяє користувачу одразу бачити вплив кількості гостей на підсумкову вартість проживання. У разі перевищення базової кількості гостей система автоматично додає відповідну доплату згідно з параметрами, заданими власником житла. Такий підхід забезпечує прозорість фінансових розрахунків і зменшує ймовірність виникнення помилок під час оформлення бронювання. Сторінку підтвердження бронювання подано на рис. 3.7.

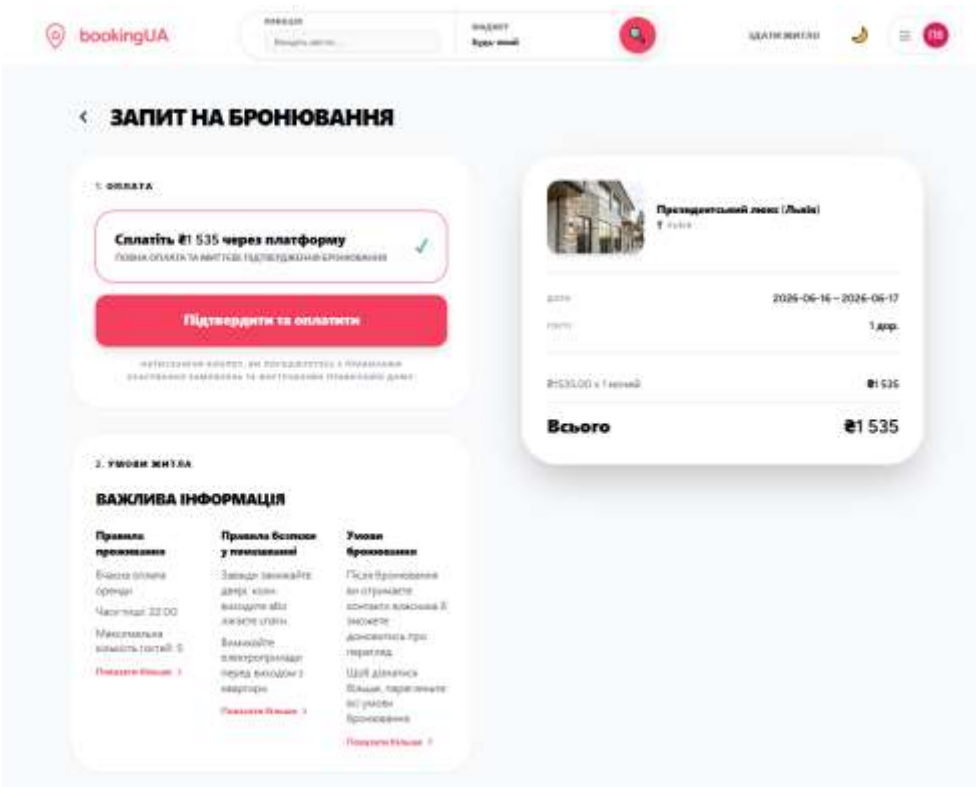


Рис. 3.7. Сторінка підтвердження бронювання

Після успішного оформлення бронювання система автоматично формує повідомлення власнику житла. У повідомленні відображається інформація про орендаря, об'єкт розміщення, період проживання, кількість гостей, загальну вартість замовлення та спосіб оплати. Такий механізм дозволяє власнику оперативно отримувати інформацію про нові бронювання та за необхідності підтримувати зв'язок з орендарем через вбудований чат, що відображено на рис. 3.8.

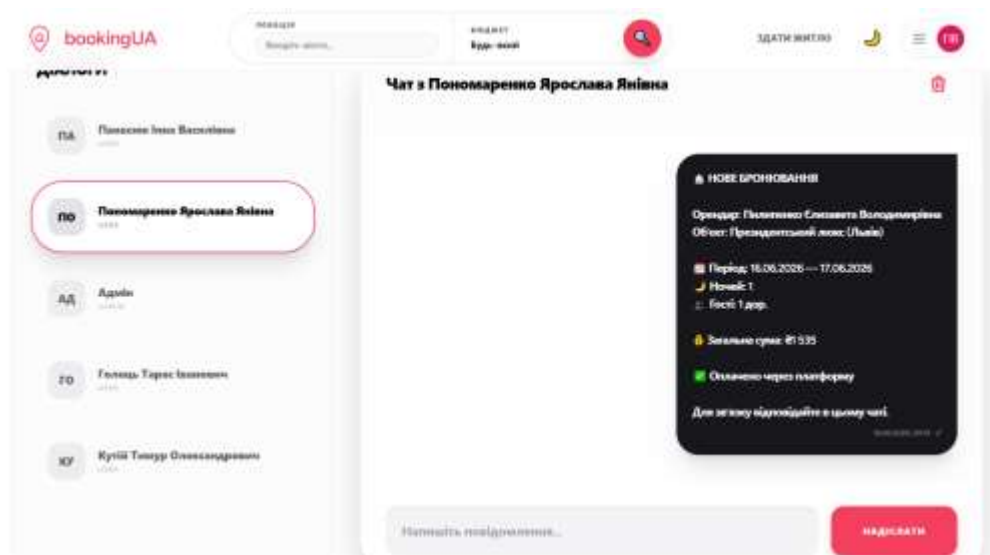


Рис. 3.8. Автоматичне повідомлення власнику про нове бронювання

Перед підтвердженням бронювання користувач може ознайомитися з правилами проживання та правилами безпеки у помешканні. Вони відображаються у модальних вікнах і містять основні умови користування житлом, обмеження, максимальну кількість гостей та рекомендації щодо безпечного перебування. Наявність таких інформаційних блоків дозволяє користувачу ознайомитися з основними умовами проживання ще до підтвердження бронювання. Це зменшує ризик непорозумінь між орендарем і власником житла та підвищує прозорість процесу бронювання. Приклади перегляду правил проживання та правил безпеки наведено на рис. 3.9 і рис. 3.10 відповідно.

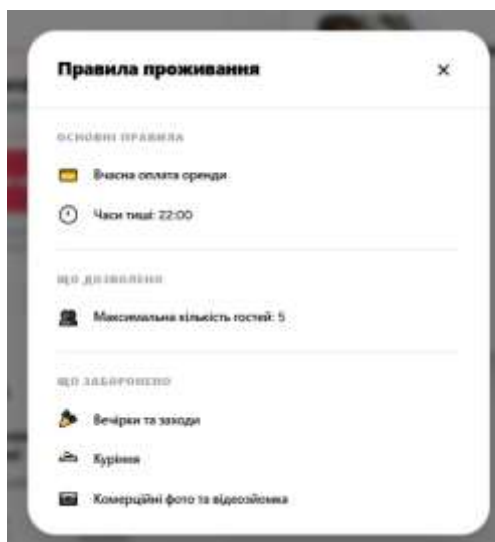


Рис. 3.9. Перегляд правил проживання об'єкта

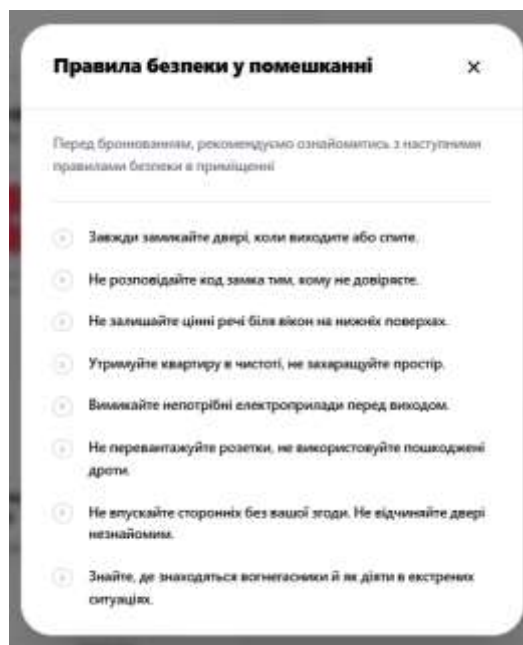


Рис. 3.10. Перегляд правил безпеки у помешканні

Після створення бронювання відповідна інформація зберігається в базі даних, а користувач може переглядати її в особистому кабінеті. У розділі «Поїздки» відображається інформація про всі здійснені бронювання із зазначенням назви об'єкта, місця розташування, періоду проживання, статусу та вартості. Для активного бронювання користувач може виконати його скасування. Після завершення періоду проживання система автоматично надає можливість залишити відгук про об'єкт розміщення, що забезпечує формування рейтингу житла та накопичення відгуків від користувачів.

Для зручності користувача бронювання подаються у вигляді окремих інформаційних блоків, що дає змогу швидко переглядати основні відомості про кожну поїздку. Статус бронювання дозволяє визначити, чи є воно активним, завершеним або скасованим. Доступні дії залежать від поточного стану бронювання та періоду проживання. Такий підхід запобігає виконанню некоректних операцій, зокрема скасуванню вже завершеної поїздки або передчасному залишенню відгуку. Відображення всієї інформації в одному розділі спрощує контроль користувачем власних бронювань і забезпечує зручний доступ до історії поїздок.

Приклад сторінки перегляду поїздок користувача та доступних дій із бронюваннями наведено на рис. 3.11.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		70

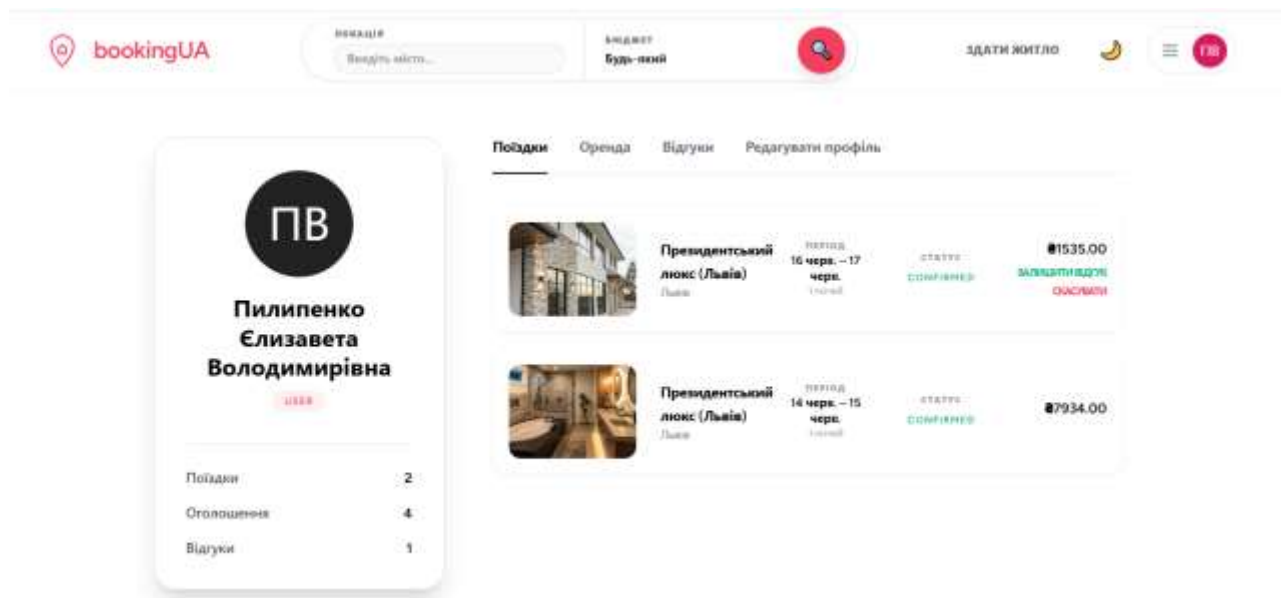


Рис. 3.11. Перегляд поїздок користувача та доступні дії з бронюваннями

Після завершення періоду проживання користувач отримує можливість залишити відгук про орендоване житло. Форма відгуку містить оцінку за п'ятизірковою шкалою та поле для текстового коментаря. Можливість залишення відгуку надається лише після завершення бронювання, що підвищує достовірність оцінювання житла. Перед збереженням відгуку система перевіряє наявність завершеного бронювання відповідного об'єкта цим користувачем. Це запобігає створенню відгуків користувачами, які фактично не проживали в обраному житлі. Форму додавання відгуку наведено на рис. 3.12.

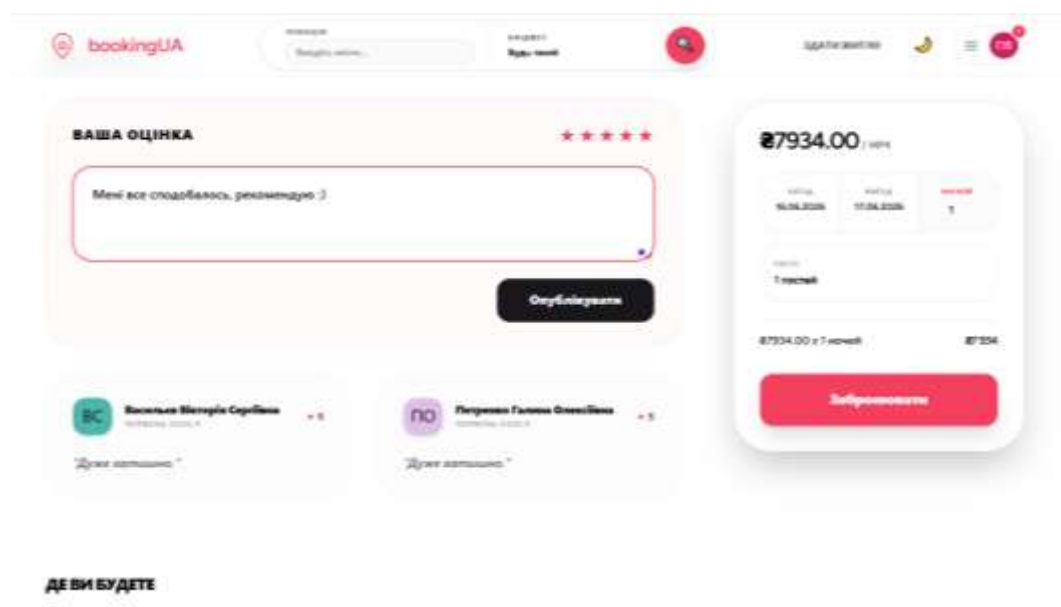


Рис. 3.12. Форма додавання відгуку після завершення проживання

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		71

Після успішного збереження відгуку система відображає повідомлення про його публікацію, а сам відгук зберігається в особистому кабінеті користувача. У вкладці «Відгуки» доступний перегляд залишених оцінок, текстових коментарів, дати публікації та інформації про відповідне житло. Перегляд залишених користувачем відгуків подано на рис. 3.13.

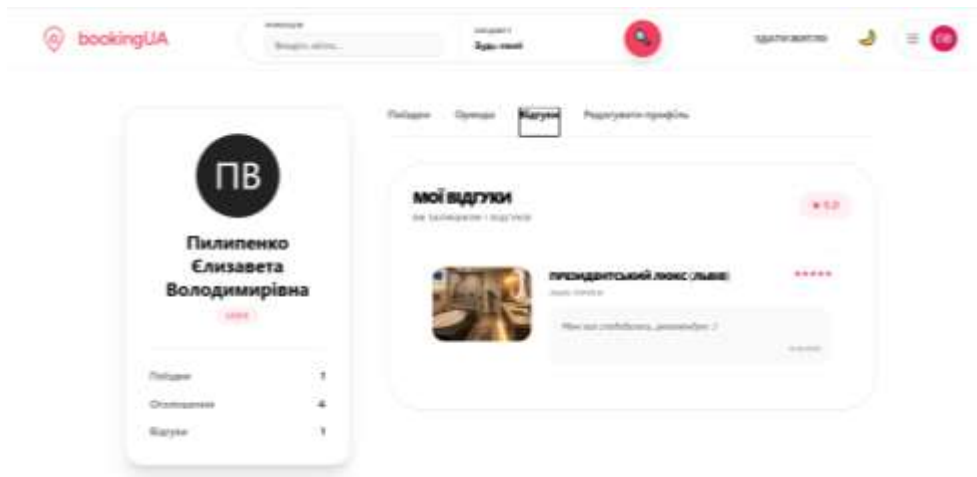


Рис. 3.13. Перегляд залишених відгуків в особистому кабінеті

Користувач може виступати не лише орендарем, а й власником житла. У вкладці «Оренда» особистого кабінету відображаються власні оголошення, їхній статус, кількість бронювань і фінансові показники. Для власника виводиться загальна сума за підтвердженими бронюваннями, комісія платформи та сума, нарахована власнику. Кабінет власника житла наведено на рис. 3.14.

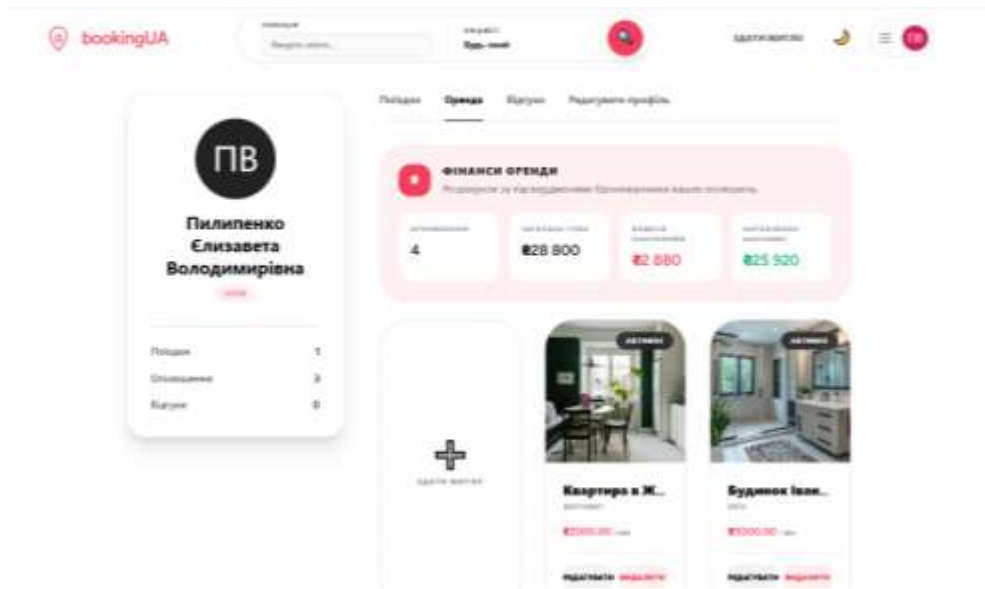


Рис. 3.14. Особистий кабінет власника житла

Створення нового оголошення виконується через спеціальну форму. Власник зазначає назву, місто, ціну за ніч, адресу, базову кількість гостей, доплату за додаткового гостя, опис житла та розташування на інтерактивній карті. Використання карти дозволяє точніше визначити місцезнаходження житла та відобразити його на сторінці оголошення. Форму створення нового оголошення подано на рис. 3.15.

Рис. 3.15. Форма створення нового оголошення

Для кожного оголошення обов'язковим є додавання не менше трьох фотографій. У формі створення відображається кількість завантажених зображень і мінімальна вимога до їх кількості. Також користувачеві показується інформація про комісійну модель платформи: після підтвердженого бронювання частина суми утримується платформою, а решта нараховується власнику.

Після заповнення форми система перевіряє наявність обов'язкових даних, кількість доданих фотографій та коректність вказаних параметрів. Якщо всі умови виконано, оголошення надсилається на перевірку адміністратору й тимчасово отримує статус очікування. Такий підхід дає змогу запобігти появі неповних або некоректних оголошень у загальному каталозі житла.

Завершальний етап створення оголошення наведено на рис. 3.16.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		73

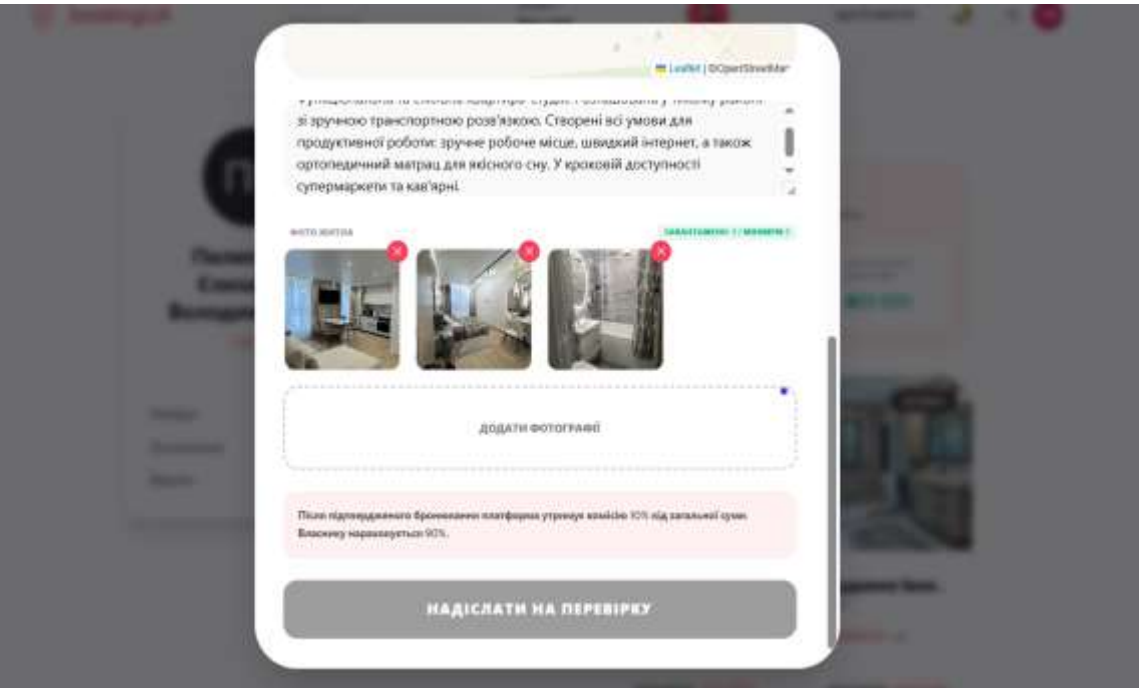


Рис. 3.16. Завантаження фотографій та завершення створення оголошення

Після створення оголошення воно надсилається на перевірку адміністратору та не стає активним автоматично. У кабінеті власника відображаються статуси оголошень, зокрема активні та неактивні. Для неактивного оголошення передбачено можливість повідомити адміністратора. Відображення статусів оголошень у кабінеті власника наведено на рис. 3.17.

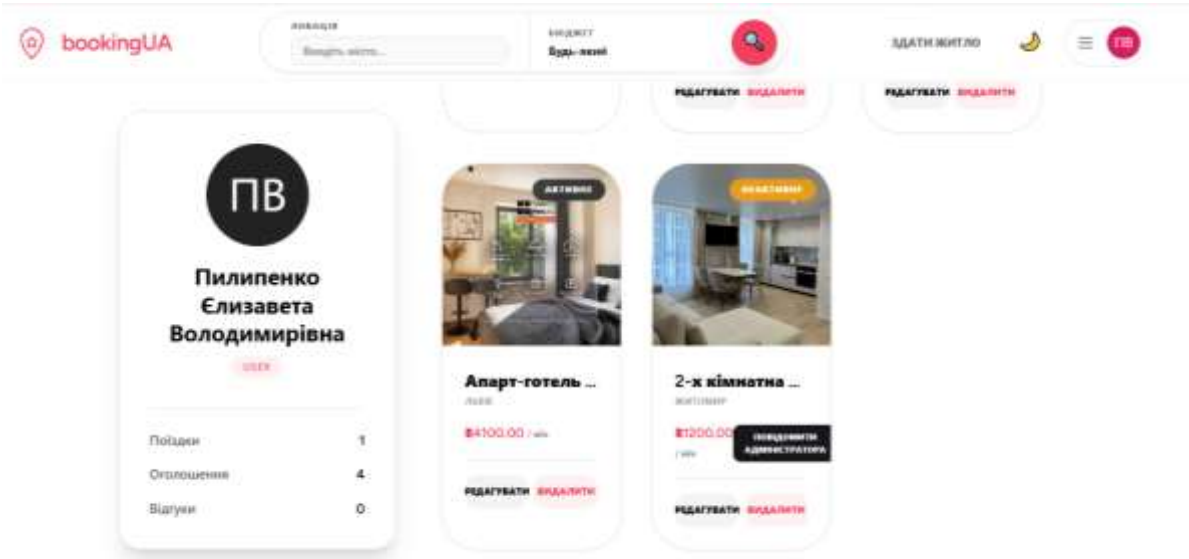


Рис. 3.17. Відображення статусів оголошень у кабінеті власника

Для зручності роботи у вебсервісі реалізовано персональне меню користувача, через яке здійснюється доступ до основних розділів системи. Залежно від ролі користувача меню містить переходи до особистого кабінету,

повідомлень, підтримки, сторінки створення оголошень та інших функцій. Також у меню відображається індикатор непрочитаних повідомлень, що дозволяє користувачу оперативно реагувати на нові звернення та відповіді. Меню навігації авторизованого користувача наведено на рис. 3.18.

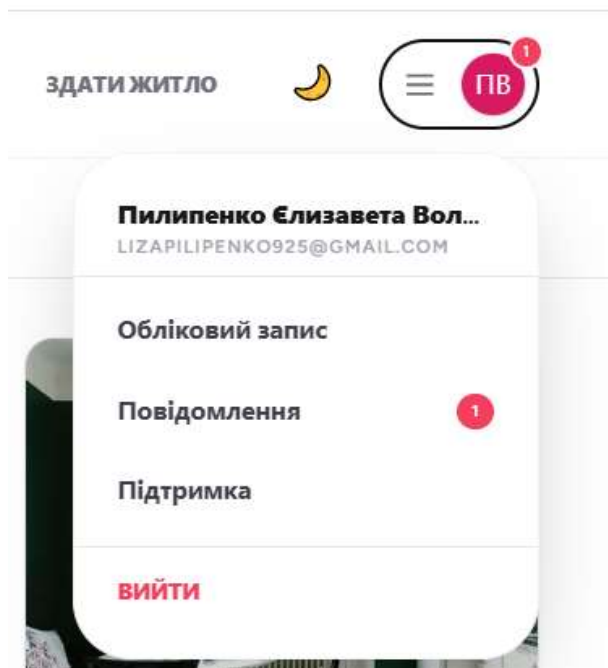


Рис. 3.18. Меню навігації авторизованого користувача

Окрім користувацької частини, у вебсервісі реалізовано адміністративну панель, призначену для контролю роботи системи, керування користувачами, оголошеннями, бронюваннями, повідомленнями та статистичними показниками. Доступ до цього розділу має лише користувач з роллю адміністратора, що забезпечує розмежування функцій між звичайними користувачами та особою, яка виконує контроль і супровід роботи вебсервісу.

Головна сторінка адміністративної панелі містить узагальнені показники роботи вебсервісу, зокрема кількість зареєстрованих користувачів, фінансові показники платформи, середній час відповіді та інструмент формування PDF-звіту. Така сторінка виконує роль інформаційної панелі, з якої адміністратор може швидко оцінити загальний стан системи та перейти до керування окремими модулями. Загальний вигляд адміністративної панелі наведено на рис. 3.19.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

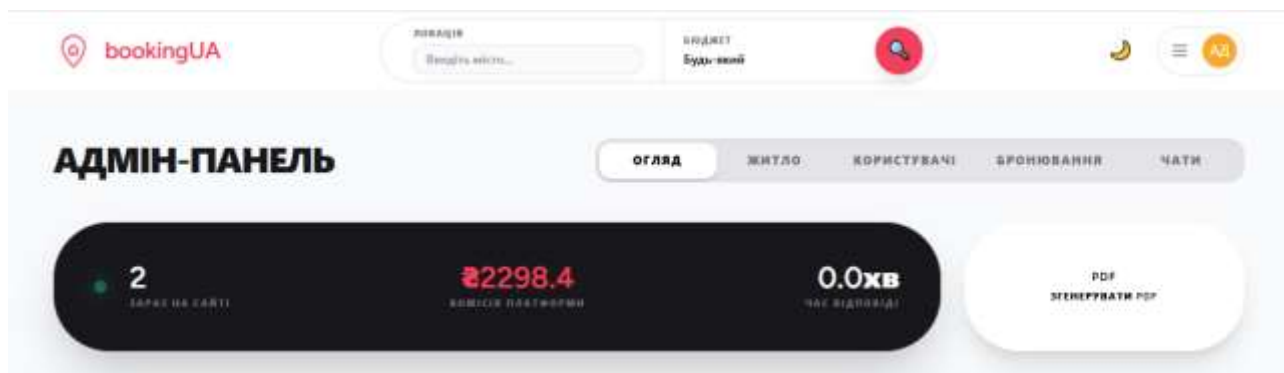


Рис. 3.19. Головна сторінка адміністративної панелі вебсервісу

Для аналізу розміщених оголошень у системі передбачено карту об'єктів житла. Вона дозволяє адміністратору переглядати географічне розташування пропозицій і визначати, у яких містах або районах зосереджена найбільша кількість активних об'єктів. Таке подання інформації є зручним для швидкої оцінки наповнення платформи та виявлення територій із недостатньою кількістю пропозицій. Карта також допомагає перевірити коректність координат, указаних власниками під час створення оголошень. Якщо об'єкт розташований некоректно, адміністратор може додатково перевірити дані оголошення та за потреби звернутися до власника житла. Географічне відображення об'єктів житла наведено на рис. 3.20.

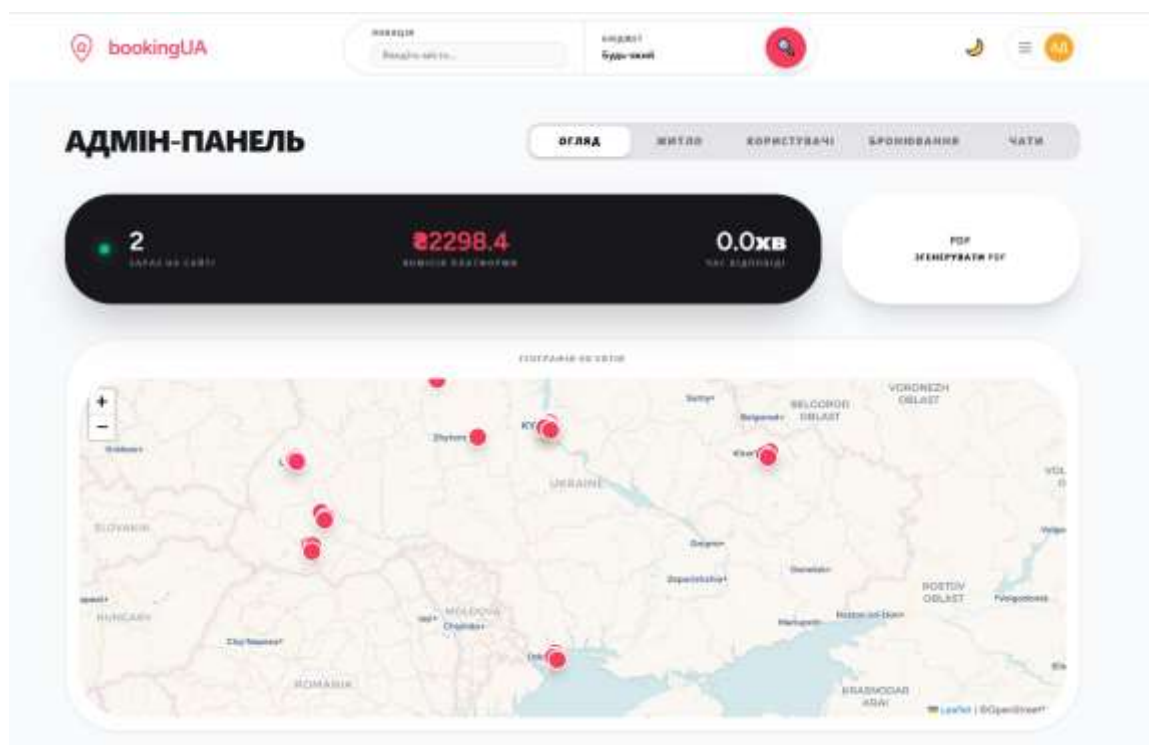


Рис. 3.20. Географічне відображення об'єктів житла

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

Окремим елементом адміністративної панелі є модуль виявлення підозрілої активності. Він використовується для звернення уваги адміністратора на оголошення або дії, які можуть потребувати додаткової перевірки. Наприклад, система може позначати об'єкти з відсутнім або занадто коротким описом, що дозволяє адміністратору швидше знаходити проблемні записи та перевіряти їх у відповідному розділі панелі. Приклад роботи модуля виявлення підозрілої активності наведено на рис. 3.21.

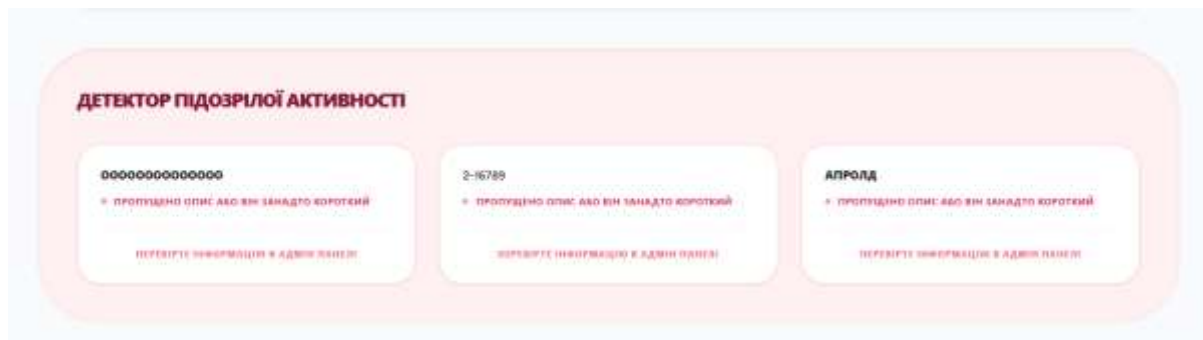


Рис. 3.21. Модуль виявлення підозрілої активності

В адміністративній частині також реалізовано аналітичні блоки, що відображають статистику роботи вебсервісу. До таких показників належать популярність міст, сезонність бронювань, активність користувачів і рейтинг популярних об'єктів житла. Окремо передбачено модуль виявлення підозрілої активності, який дає змогу адміністратору звернути увагу на нетипові дії користувачів або об'єктів. Аналітичні блоки адміністративної панелі подано на рис. 3.22.



Рис. 3.22. Аналітичні блоки адміністративної панелі

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

Рейтинг популярних об'єктів житла формується на основі кількості бронювань і дає змогу визначити оголошення, які мають найбільший попит серед користувачів. Такий інструмент може використовуватися адміністратором для оцінювання активності сервісу та якості розміщених пропозицій. Приклад рейтингу популярних об'єктів наведено на рис. 3.23.

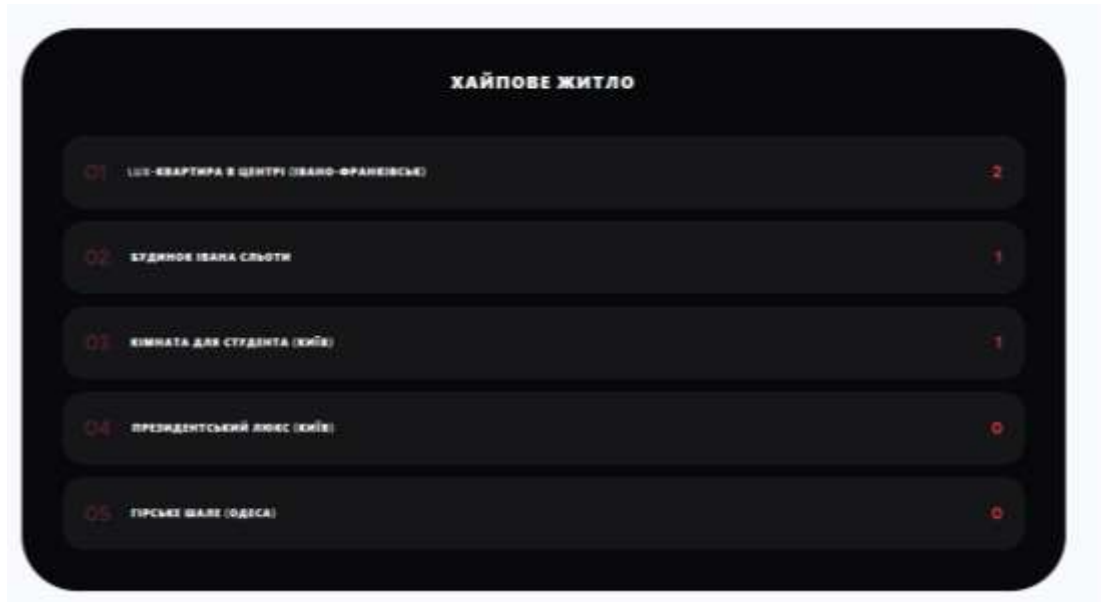


Рис. 3.23. Рейтинг популярних об'єктів житла

Для фінансового аналізу в адміністративній частині передбачено можливість автоматичного формування PDF-звіту. Такий звіт містить узагальнені фінансові показники платформи, зокрема прогноз доходу, кількість успішних бронювань, середній фінансовий чек, показники повторного використання сервісу та реєстр фінансових операцій. Це дає змогу використовувати адміністративну панель не лише для поточного керування системою, а й для підготовки аналітичних матеріалів щодо ефективності роботи платформи.

Автоматичне формування звіту зменшує потребу в ручному опрацюванні даних, а їх структуроване подання спрощує перегляд і аналіз. Формат PDF забезпечує однакове відображення документа на різних пристроях і дає змогу зберігати звіти для подальшого використання. Приклад автоматично сформованого PDF-звіту наведено на рис. 3.24.

BookingUA: Фінансово-аналітичний звіт

Дата формування: 17.06.2026, 01:47:47
Призначення: Деталізований звіт для фінансового аудиту та аналізу ефективності

Financial Metric & Description	Value & Financial Context
Прогноз загального доходу на період	2 296,4 UAH — це потенційний грошовий оборот на основі всіх активних та підтверджених бронювань на наступні 30 календарних днів.
Ефективність реалізації (успішні заїзди)	9 одиниць — загальна кількість заїздів, які були фактично підтверджені та оплачені клієнтами.
Рівень лояльності бази (Retention)	3 клієнтів — кількість унікальних користувачів, які скористалися сервісом 2 та більше разів за весь час (показник стабільності).
Середній фінансовий чек (чек на транзакцію)	135.20 UAH — середня сума, яку приносить одне успішне бронювання до бюджету платформи.
Швидкість реагування орендодавців	0.0 хв. — середньостатистичний час, який витрачає власник житла на відповідь у внутрішньому чаті.

Рис. 3.24. Автоматично сформований фінансово-аналітичний PDF-звіт

Для керування оголошеннями передбачено розділ «Житло». У ньому адміністратор може переглядати всі об’єкти, виконувати пошук, аналізувати статус оголошень, переглядати детальну інформацію, активувати, деактивувати або видаляти записи. Нові оголошення після створення власником отримують статус очікування перевірки й не відображаються у відкритій частині системи до підтвердження адміністратором. Розділ керування житлом наведено на рис. 3.25.

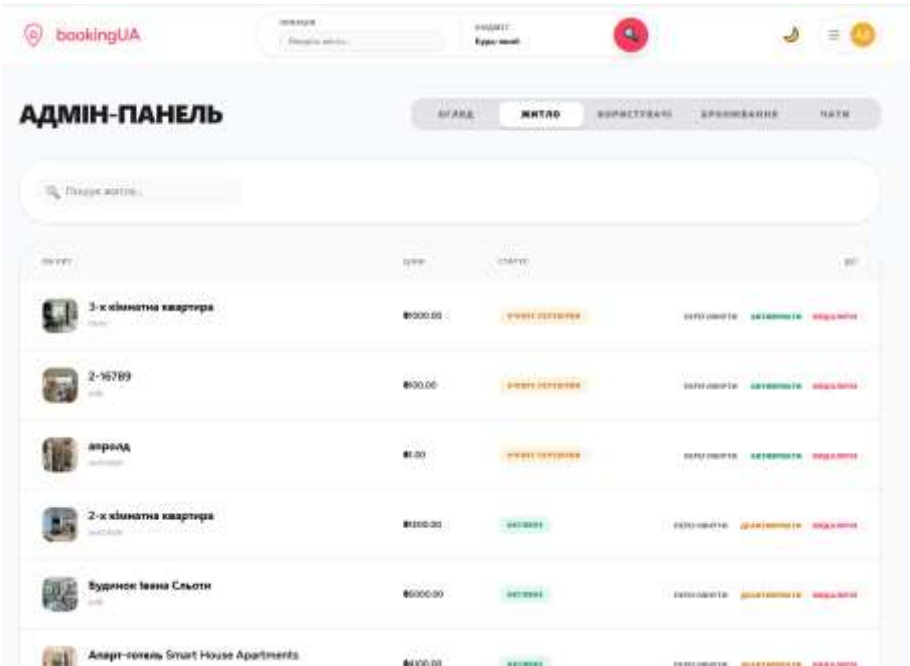


Рис. 3.25. Розділ керування об’єктами житла

Під час модерації адміністратор може активувати оголошення або деактивувати його із зазначенням причини. У разі активації власник отримує повідомлення про успішне підтвердження оголошення. Якщо оголошення потребує уточнення або не відповідає вимогам, адміністратор може надіслати повідомлення користувачу через внутрішній чат. Такий підхід забезпечує контроль якості оголошень і створює канал зворотного зв'язку між адміністрацією та власником житла.

Для керування користувачами реалізовано окремий розділ, у якому адміністратор може переглядати список облікових записів, виконувати пошук, переглядати роль і статус користувача, редагувати дані або видаляти запис. Цей функціонал необхідний для підтримки актуальності користувацької бази та контролю доступу до системи. Розділ керування користувачами наведено на рис. 3.26.

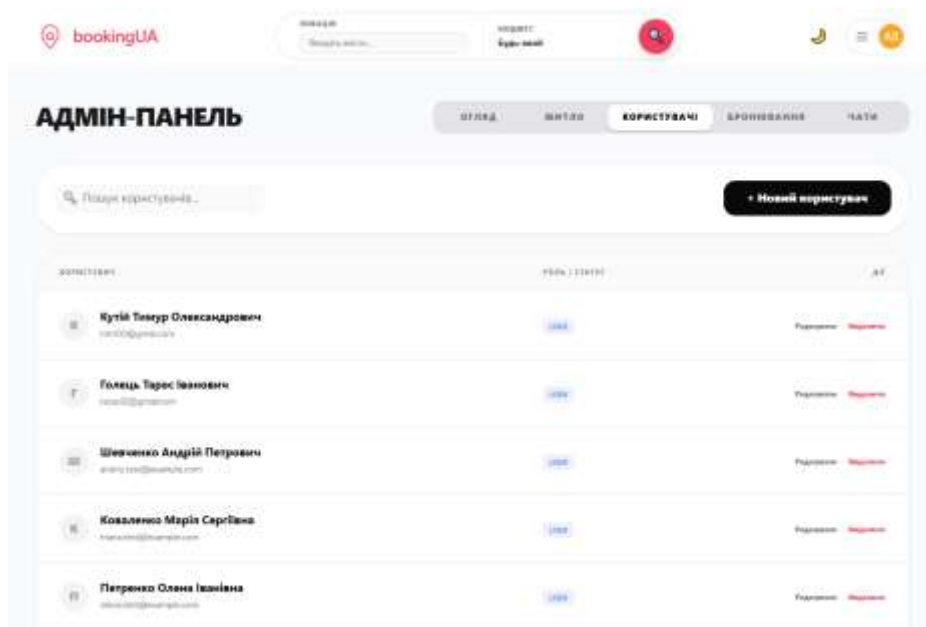


Рис. 3.26. Розділ керування користувачами

Розділ керування бронюваннями дає змогу адміністратору переглядати всі створені бронювання, зокрема ім'я гостя, назву житла, період проживання, суму, статус і доступні дії. У системі відображаються як підтверджені, так і скасовані бронювання. Для активних бронювань передбачено можливість скасування, тоді як для скасованих записів дії недоступні. Розділ керування бронюваннями подано на рис. 3.27.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІПЗ	Арк.
		Локтікова Т.М.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

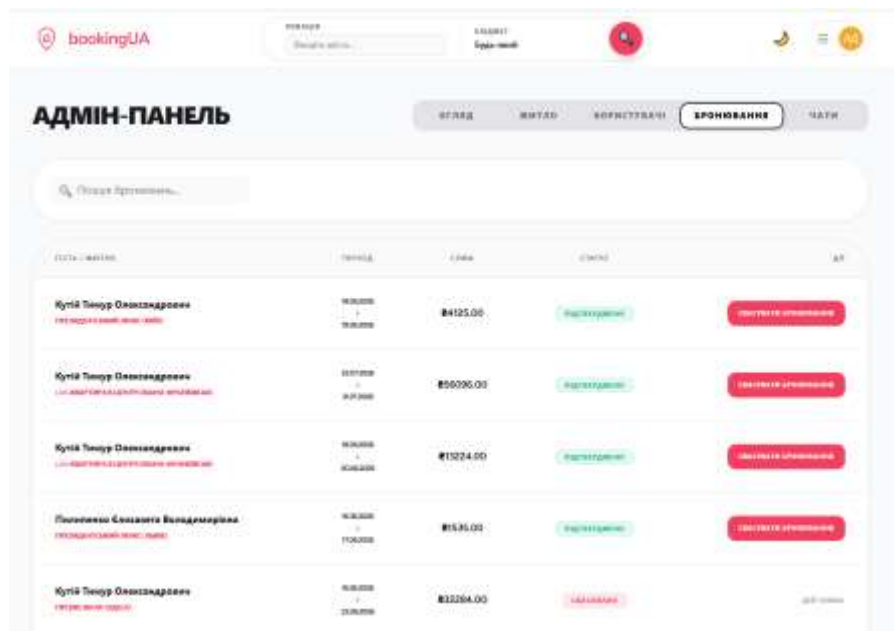


Рис. 3.27. Розділ керування бронюваннями

В адміністративній частині також реалізовано модуль чатів. Він використовується для опрацювання звернень користувачів, перевірки нових оголошень і комунікації з власниками житла. У списку чатів адміністратор може бачити користувачів, які очікують відповіді або мають оголошення на перевірку. Після вибору користувача відкривається історія повідомлень, пов'язаних із відповідними об'єктами житла. Модуль чатів адміністратора наведено на рис. 3.28.

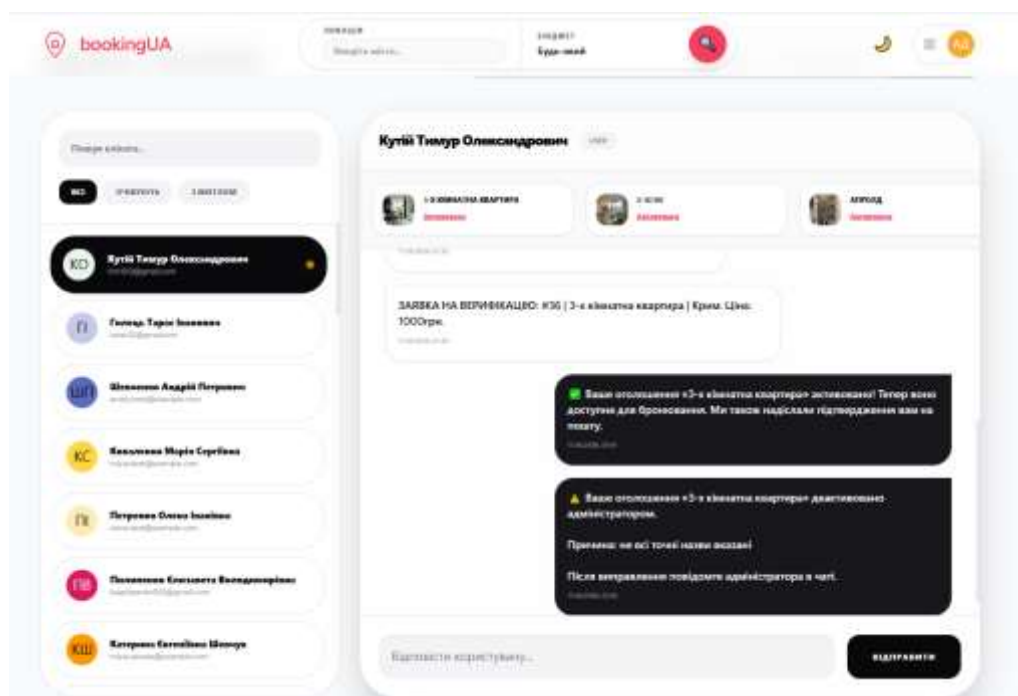


Рис. 3.28. Модуль чатів адміністратора

Через чат адміністратор може повідомити користувача про результати перевірки оголошення, надати рекомендації щодо виправлення даних або підтвердити активацію об'єкта. У повідомленнях відображається дата й час надсилання, а також інформація про об'єкт, якого стосується звернення. Це дозволяє зберігати історію взаємодії між адміністрацією та користувачем.

Отже, у межах реалізації інтерфейсу вебсервісу було створено як користувацьку, так і адміністративну частини системи. Користувацька частина забезпечує пошук житла, бронювання, умовну оплату, відгуки, обране, створення оголошень і перегляд фінансових показників власника. Адміністративна частина забезпечує керування користувачами, оголошеннями, бронюваннями, повідомленнями, статистикою, картою об'єктів, популярністю житла, PDF-звітами та виявленням підозрілої активності. Реалізовані функціональні модулі забезпечують повний цикл роботи вебсервісу та створюють основу для подальшого тестування системи.

3.3. Тестування роботи вебсервісу

Після завершення розробки вебсервісу було проведено комплексне тестування реалізованого функціоналу. Основною метою тестування стала перевірка коректності роботи системи бронювання житла, відповідності програмного продукту визначеним вимогам та забезпечення стабільної роботи всіх компонентів вебзастосунку.

У процесі тестування перевірялася робота користувацької та адміністративної частин системи. Особливу увагу приділено процесам пошуку житла, створення та скасування бронювань, перевірці доступності обраних дат, розрахунку вартості проживання, роботі системи повідомлень, відгуків та керуванню оголошеннями. Також виконано перевірку механізмів автентифікації та авторизації користувачів відповідно до їхніх ролей.

Тестування проводилося шляхом виконання типових сценаріїв використання вебсервісу відповідно до загальних принципів тестування програмних систем [19, 32]. Для кожного сценарію визначалися початкові умови,

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

послідовність дій користувача та очікуваний результат. Отримані результати порівнювалися з передбаченою логікою роботи системи.

Результати проведеного тестування наведено в табл. 3.1.

Таблиця 3.1

Результати функціонального тестування

№	Тестовий сценарій	Очікуваний результат	Результат	Статус
1.	Реєстрація нового користувача	Створення облікового запису та автоматичний вхід до системи	Відповідає очікуваному	Успішно
2.	Авторизація користувача	Вхід до особистого кабінету після введення коректних даних	Відповідає очікуваному	Успішно
3.	Пошук житла за містом	Відображення оголошень відповідно до введеного міста	Відповідає очікуваному	Успішно
4.	Фільтрація за бюджетом	Відображення житла у вибраному ціновому діапазоні	Відповідає очікуваному	Успішно
5.	Перегляд детальної сторінки житла	Відображення опису, фотографій, відгуків та умов проживання	Відповідає очікуваному	Успішно
6.	Створення бронювання на вільні дати	Формування нового бронювання та розрахунок вартості	Відповідає очікуваному	Успішно
7.	Спроба бронювання зайнятих дат	Відображення повідомлення про недоступність дат	Відповідає очікуваному	Успішно
8.	Бронювання суміжних дат	Бронювання успішно створюється	Відповідає очікуваному	Успішно
9.	Розрахунок доплати за додаткових гостей	Загальна сума автоматично перераховується	Відповідає очікуваному	Успішно
10.	Скасування бронювання користувачем	Статус бронювання змінюється на «Скасовано»	Відповідає очікуваному	Успішно

11.	Автоматичне створення чату після бронювання	Створюється діалог між орендарем та власником житла	Відповідає очікуваному	Успішно
12.	Надсилання повідомлення у чаті	Повідомлення відображається обом учасникам діалогу	Відповідає очікуваному	Успішно
13.	Створення нового оголошення	Оголошення успішно додається до системи	Відповідає очікуваному	Успішно
14.	Перевірка оголошення адміністратором	Зміна статусу оголошення після модерації	Відповідає очікуваному	Успішно
15.	Залишення відгуку після завершення проживання	Відгук додається до сторінки житла	Відповідає очікуваному	Успішно
16.	Формування PDF-звіту адміністратором	Генерується файл зі статистикою платформи	Відповідає очікуваному	Успішно
17.	Відображення об'єктів на карті	На карті відображаються доступні оголошення	Відповідає очікуваному	Успішно
18.	Перемикання світлої та темної теми	Інтерфейс коректно змінює оформлення	Відповідає очікуваному	Успішно

За результатами проведеного функціонального тестування встановлено, що основні модулі вебсервісу працюють відповідно до визначеної логіки. Система коректно виконує реєстрацію та авторизацію користувачів, забезпечує пошук і фільтрацію житла, відображає детальну інформацію про об'єкти та виконує автоматичний розрахунок вартості бронювання.

Окремо перевірено процес створення бронювання. Під час тестування система правильно визначала кількість ночей, враховувала доплату за додаткових гостей і не дозволяла створювати бронювання на вже зайняті дати.

Водночас суміжні бронювання, коли дата виїзду одного користувача збігається з датою заїзду іншого, оброблялися коректно та дозволялися системою.

Також перевірено роботу особистого кабінету користувача та власника житла. У кабінеті користувача правильно відображалися поїздки, статуси бронювань і можливість залишення відгуку після завершення проживання. У кабінеті власника коректно відображалися створені оголошення, їхні статуси, фінансові показники та повідомлення про нові бронювання.

Під час тестування адміністративної частини підтверджено працездатність модулів керування оголошеннями, користувачами, бронюваннями та повідомленнями. Адміністратор має змогу переглядати статистичні показники системи, активувати або деактивувати оголошення, контролювати бронювання та формувати PDF-звіт. Отримані результати свідчать про коректну роботу основних адміністративних функцій.

Крім функціональних можливостей, перевірено коректність відображення інтерфейсу у світлій і темній темі. Перемикання теми виконувалося без порушення структури сторінок, а основні елементи інтерфейсу залишалися доступними для користувача. За результатами тестування критичних помилок у роботі вебсервісу не виявлено.

Для додаткової технічної перевірки клієнтської частини було виконано створення виробничої збірки у середовищі Ubuntu за допомогою команди `./vendor/bin/sail npm run build`. У результаті Vite успішно сформував оптимізовані файли застосунку в каталозі `public/build`. Під час збірки було виведено попередження щодо розміру окремих фрагментів JavaScript, однак критичних помилок не виявлено.

Успішне створення виробничої збірки підтверджує правильність налаштування середовища розробки, коректне підключення залежностей і відсутність помилок, які перешкоджають компіляції клієнтської частини. Сформовані файли можуть бути використані для подальшого розгортання вебсервісу в робочому середовищі.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		85

Під час аналізу результатів збірки встановлено, що Vite сформував окремі CSS- та JavaScript-файли для сторінок і компонентів вебсервісу. Наявність згенерованого файлу manifest.json підтверджує коректне створення зв'язків між початковими ресурсами та зібраними файлами. Це дає змогу Laravel правильно підключати оптимізовані клієнтські ресурси під час подальшого запуску вебсервісу. Крім того, успішне завершення процесу збирання підтверджує сумісність використаних бібліотек і відсутність конфліктів між залежностями проєкту. Отриманий результат свідчить про готовність клієнтської частини до розгортання та використання в робочому середовищі. Результат виконання команди наведено на рис. 3.29.

```
> build
> vite build

vite v6.4.1 building for production...
1129 modules transformed.
public/build/assets/layers-BWBAp2CZ.png 0.70 kB
public/build/assets/layers-2x-Bpkbi35X.png 1.26 kB
public/build/assets/marker-icon-hN30_KVU.png 1.47 kB
public/build/manifest.json 13.80 kB
public/build/assets/MapPicker-IR_xTkdI.css 0.07 kB
public/build/assets/Edit-B3UJ2twI.css 0.14 kB
public/build/assets/StatsOverview-CvBuMF_1.css 0.15 kB
public/build/assets/Checkout-Ba3f3ieC.css 0.20 kB
public/build/assets/Index-BZELHtkB.css 0.33 kB
public/build/assets/MyBookings-7KxbGS6Z.css 0.33 kB
public/build/assets/AppHeader-BIAq_vR.css 0.33 kB
public/build/assets/Show-BueoYH3B.css 0.54 kB
public/build/assets/Dashboard-DK7Mda2.css 0.60 kB
public/build/assets/leaflet-CG6Damiu.css 10.51 kB
public/build/assets/app-BaVd3Av8.css 89.12 kB
public/build/assets/_plugin-vue_export-helper-DIAUqK2U.js 0.09 kB
public/build/assets/InputLabel-C9lrV-AK.js 0.33 kB
public/build/assets/InputError-D0UHksQa.js 0.37 kB
public/build/assets/PrimaryButton-DuD3viol.js 0.55 kB
public/build/assets/TextInput-Ch-ingER.js 0.57 kB
public/build/assets/Dashboard-CfBlqh6b.js 0.64 kB
public/build/assets/Analytics-SmmDMrTf.js 0.89 kB
public/build/assets/GuestLayout-D4V7v6lC.js 1.03 kB
public/build/assets/ConfirmPassword-CcDe9D9H.js 1.38 kB
public/build/assets/ForgotPassword-oNBkzCU.js 1.56 kB
public/build/assets/VerifyEmail-CpfNcluv.js 1.56 kB
public/build/assets/MapPicker-DpDkjKe0.js 1.87 kB
public/build/assets/Blocked-SNRiFaXv.js 1.88 kB
public/build/assets/ResetPassword-C4bFtgfQ.js 2.12 kB
public/build/assets/AppFooter-CflaC-Lk.js 3.30 kB
public/build/assets/UpdateProfileInformationForm-BYBLnIPW.js 3.44 kB
public/build/assets/MyReviews-D6COKRzS.js 3.97 kB
public/build/assets/UpdatePasswordForm-Do4_bCCg.js 4.09 kB
public/build/assets/Login-rbRwXuka.js 4.88 kB
public/build/assets/MyBookings-C_Vz21vR.js 5.17 kB
public/build/assets/AuthenticatedLayout-Dofng3Dx.js 5.41 kB
public/build/assets/Edit-FJTULML.js 5.49 kB
public/build/assets/Register-NeeV_QHx.js 5.55 kB
public/build/assets/Index-CXJQJzq7.js 8.50 kB
public/build/assets/Welcome-DSNv7kqH.js 8.92 kB
public/build/assets/AppHeader-BzoTrMXF.js 12.06 kB
public/build/assets/MyProperties-BzMaBy7-.js 13.84 kB
public/build/assets/Checkout-BRHcf02D.js 17.08 kB
public/build/assets/purify-es-BgtPMW3.js 22.77 kB
gzip: 1.80 kB
gzip: 0.08 kB
gzip: 0.11 kB
gzip: 0.13 kB
gzip: 0.13 kB
gzip: 0.15 kB
gzip: 0.15 kB
gzip: 0.15 kB
gzip: 0.21 kB
gzip: 0.24 kB
gzip: 2.65 kB
gzip: 14.93 kB
gzip: 0.10 kB
gzip: 0.26 kB
gzip: 0.29 kB
gzip: 0.38 kB
gzip: 0.39 kB
gzip: 0.42 kB
gzip: 0.60 kB
gzip: 0.62 kB
gzip: 0.78 kB
gzip: 0.88 kB
gzip: 0.90 kB
gzip: 1.16 kB
gzip: 0.95 kB
gzip: 0.86 kB
gzip: 1.54 kB
gzip: 1.67 kB
gzip: 1.82 kB
gzip: 1.56 kB
gzip: 2.07 kB
gzip: 2.32 kB
gzip: 2.16 kB
gzip: 2.38 kB
gzip: 2.02 kB
gzip: 3.49 kB
gzip: 3.74 kB
gzip: 4.13 kB
gzip: 4.56 kB
gzip: 5.40 kB
gzip: 8.79 kB
```

Рис. 3.29. Результат створення виробничої збірки вебсервісу

Проведене тестування підтвердило, що основні функціональні модулі вебсервісу працюють відповідно до визначених вимог. Система коректно обробляє пошук житла, створення бронювання, перевірку перетину дат, розрахунок вартості, умовну оплату, скасування бронювання, залишення відгуків, роботу повідомлень та адміністративне керування оголошеннями.

Результати створення виробничої збірки засвідчили, що клієнтська частина вебсервісу успішно збирається засобами Vite. Попередження щодо розміру окремих JavaScript-фрагментів не є критичною помилкою і може бути усунене під час подальшої оптимізації шляхом поділу коду на менші частини. Отже, вебсервіс є працездатним і готовим до подальшого використання та вдосконалення.

Висновки до розділу 3

У третьому розділі описано порядок установлення та налаштування вебсервісу. Розглянуто встановлення серверних і клієнтських залежностей, створення конфігураційного файлу, налаштування підключення до PostgreSQL, запуск контейнерного середовища за допомогою Docker і Laravel Sail, виконання міграцій та запуск Vite.

Наведено структуру користувацького інтерфейсу та порядок взаємодії з вебсервісом. Описано роботу головної сторінки, пошуку й фільтрації житла, перегляду оголошень, бронювання, особистого профілю, обраного, повідомлень і засобів керування власними об'єктами. Також розглянуто можливості адміністративної частини вебсервісу.

Під час функціонального тестування перевірено основні сценарії роботи системи, правильність опрацювання введених даних, відображення інформації та виконання операцій відповідно до ролі користувача. Результати перевірки підтвердили працездатність основних функцій вебсервісу та коректність взаємодії його компонентів.

Отже, у розділі визначено послідовність розгортання системи, описано порядок роботи з її інтерфейсом і перевірено функціонування вебсервісу.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		87

ВИСНОВКИ

У межах випускної кваліфікаційної роботи бакалавра спроектовано та реалізовано вебсервіс для бронювання житла. Розроблена система забезпечує пошук і фільтрацію оголошень, перегляд інформації про житло, створення та скасування бронювань, перевірку доступності дат, розрахунок вартості проживання, роботу з обраним, відгуками та повідомленнями, а також створення і модерацію оголошень.

У першому розділі проаналізовано предметну область короткострокового бронювання житла та розглянуто наявні програмні аналоги. За результатами порівняння визначено основні функціональні вимоги до вебсервісу, обрано клієнт-серверну архітектуру та обґрунтовано використання програмних засобів для реалізації системи.

У другому розділі виконано проектування вебсервісу. Визначено ролі користувачів і доступні їм варіанти використання, розроблено об'єктно-орієнтовану структуру та структуру бази даних, описано основні алгоритми й бізнес-правила. Реалізовано створення і скасування бронювань, перевірку перетину дат, розрахунок вартості, роботу з оголошеннями, фотографіями, обраним, відгуками, повідомленнями та розмежуванням прав доступу.

У третьому розділі наведено порядок установаження й налаштування вебсервісу, описано користувацький та адміністративний інтерфейси, а також послідовність взаємодії із системою. Проведено функціональне тестування основних сценаріїв роботи та перевірено коректність виконання операцій.

Серверну частину вебсервісу реалізовано мовою PHP із використанням Laravel, клієнтську — на основі Vue.js 3, Inertia.js і Tailwind CSS; для зберігання даних, збирання ресурсів, передавання подій і контейнеризації використано PostgreSQL, Vite, Laravel Reverb, Docker та Laravel Sail. Тестування підтвердило працездатність системи та її відповідність функціональним вимогам, тому мету роботи досягнуто.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		88

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пилипенко Є.В., Локтікова Т.М., Кушнір Н.О. Сучасний стан і тенденції розвитку сервісів для бронювання житла: тези VIII Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 02–03 грудня 2025 р. Житомир: Житомирська політехніка, 2025. С. 59–60. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2026/01/povnyj-tekst-1.pdf> (дата звернення: 16.06.2026).
2. Пилипенко Є.В., Локтікова Т.М., Кушнір Н.О. Дослідження принципів побудови та проєктування вебсервісу для бронювання житла: тези XVI Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології - 2026», м. Житомир, 02–03 квітня 2026 р. Житомир: Житомирська політехніка, 2026. С. 55–56. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2026/06/povnyj-tekst.pdf> (дата звернення: 16.06.2026).
3. Doba.ua. URL: <https://doba.ua/ukr/> (дата звернення: 16.06.2026).
4. Dobovo.com. URL: <https://www.dobovo.com/ua/> (дата звернення: 16.06.2026).
5. Airbnb. URL: <https://www.airbnb.com/> (дата звернення: 16.06.2026).
6. MDN Web Docs. Client-server overview. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/First_steps/Client-Server_overview (дата звернення: 18.06.2026).
7. Gandhi R., Richards M., Ford N. Head First Software Architecture. Sebastopol: O'Reilly Media, 2024. 486 p.
8. Vue.js Documentation. URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 18.06.2026).
9. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 18.06.2026).
10. Laravel Documentation. URL: <https://laravel.com/docs> (дата звернення: 16.06.2026).
11. Stauffer M. Laravel: Up & Running. 3rd ed. Sebastopol: O'Reilly Media, 2023. 572 p.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				89
Змн.	Арк.	№ докум.	Підпис	Дата		

12. Inertia.js Documentation. URL: <https://inertiajs.com/> (дата звернення: 18.06.2026).
13. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/> (дата звернення: 18.06.2026).
14. PostgreSQL 18 Documentation. Chapter 5. Data Definition. URL: <https://www.postgresql.org/docs/current/ddl.html> (дата звернення: 19.06.2026).
15. Docker Documentation. URL: <https://docs.docker.com/> (дата звернення: 19.06.2026).
16. Laravel Sail Documentation. URL: <https://laravel.com/docs/sail> (дата звернення: 19.06.2026).
17. Vite Documentation. URL: <https://vite.dev/guide/> (дата звернення: 19.06.2026).
18. ISO/IEC 25010:2023. Systems and software engineering. Systems and software Quality Requirements and Evaluation. Product quality model. Geneva: International Organization for Standardization, 2023.
19. IEEE Computer Society. Guide to the Software Engineering Body of Knowledge (SWEBOK Guide V4.0). 2024. URL: <https://www.computer.org/education/body-of-knowledge/software-engineering/v4> (дата звернення: 19.06.2026).
20. PHP Documentation. URL: <https://www.php.net/docs.php> (дата звернення: 19.06.2026).
21. Duckett J. PHP & MySQL: Server-side Web Development. Indianapolis: Wiley, 2022. 672 p.
22. Quinten J. Building Real-World Web Applications with Vue.js 3. Birmingham: Packt Publishing, 2024. 318 p.
23. Frain B. Responsive Web Design with HTML5 and CSS. 4th ed. Birmingham: Packt Publishing, 2022. 498 p.
24. Git Documentation. URL: <https://git-scm.com/doc> (дата звернення: 19.06.2026).
25. Composer Documentation. URL: <https://getcomposer.org/doc/> (дата звернення: 19.06.2026).
26. npm Documentation. URL: <https://docs.npmjs.com/> (дата звернення: 19.06.2026).

		Пилипенко Є.В.			ІІЗ.КР.Б-121-26-ІІЗ	Арк.
		Локтікова Т.М.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

27. Node.js Documentation. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 19.06.2026).
28. PlantUML Documentation. URL: <https://plantuml.com/> (дата звернення: 19.06.2026).
29. OWASP Top Ten Web Application Security Risks. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 19.06.2026).
30. MDN Web Docs. HTTP response status codes. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Status> (дата звернення: 19.06.2026).
31. Laravel Documentation. Validation. URL: <https://laravel.com/docs/13.x/validation> (дата звернення: 19.06.2026).
32. ISO/IEC/IEEE 29119-1:2022. Software and systems engineering. Software testing. Part 1: General concepts. Geneva: International Organization for Standardization, 2022.

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		91

ДОДАТКИ

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		92

Лістинг А.1 – Валідація даних бронювання

```
$validated = $request->validate([
    'property_id' => 'required|exists:properties,id',
    'check_in' => 'required|date',
    'check_out' => 'required|date',
    'total_price' => 'nullable|numeric',
    'adults' => 'required|integer|min:1',
    'children' => 'nullable|integer|min:0',
    'payment_type' => 'required|in:full',
]);
```

Лістинг А.2 – Перевірка коректності дат бронювання

```
$now = Carbon::now();

$checkInDate = Carbon::parse($validated['check_in'])->startOfDay();
$checkOutDate = Carbon::parse($validated['check_out'])->startOfDay();

$today = $now->copy()->startOfDay();

if ($checkInDate->lt($today)) {
    return redirect()->back()->withErrors([
        'check_in' => 'Не можна забронювати житло на минулу дату.',
    ])->withInput();
}

if ($now->hour >= 22 && $checkInDate->equalTo($today)) {
    return redirect()->back()->withErrors([
        'check_in' => 'Після 22:00 бронювання на поточну дату недоступне.
Оберіть дату заїзду, починаючи з наступного дня.',
    ])->withInput();
}

if ($checkOutDate->lte($checkInDate)) {
    return redirect()->back()->withErrors([
        'check_out' => 'Дата виїзду повинна бути пізнішою за дату заїзду.',
    ]);
}
```

Лістинг А.3 – Перевірка перетину дат бронювання

```
$hasOverlappingBooking = Booking::where('property_id', $property->id)
    ->where('status', '!=', 'cancelled')
    ->where('start_date', '<', $checkOutDate->toDateString())
    ->where('end_date', '>', $checkInDate->toDateString())
    ->exists();

if ($hasOverlappingBooking) {
    return redirect()->back()->withErrors([
        'check_in' => 'Це житло вже заброньоване на вибрані дати.',
    ])->withInput();
}
```

Лістинг А.4 – Розрахунок вартості бронювання

```
$nights = max(1, $checkInDate->diffInDays($checkOutDate));

$adults = (int) $validated['adults'];
$children = (int) ($validated['children'] ?? 0);
$totalGuests = $adults + $children;

$basePricePerNight = (float) $property->price_per_night;
$baseGuests = max(1, (int) ($property->base_guests_count ?: 1));

$extraGuestPrice = (float) ($property->extra_guest_price ?? 200);

if ($extraGuestPrice <= 0) {
    $extraGuestPrice = 200;
}

$extraGuestsCount = max(0, $totalGuests - $baseGuests);

$baseTotalPrice = $basePricePerNight * $nights;
$extraGuestsTotalPrice = $extraGuestsCount * $extraGuestPrice * $nights;

$calculatedTotalPrice = $baseTotalPrice + $extraGuestsTotalPrice;

$tenantPaidAmount = round($calculatedTotalPrice, 2);
$hostAmount = round($calculatedTotalPrice * 0.90, 2);
```

Лістинг А.5 – Створення бронювання

```
$booking = Booking::create([
    'user_id' => Auth::id(),
    'property_id' => $property->id,
    'start_date' => $validated['check_in'],
    'end_date' => $validated['check_out'],
    'adults' => $adults,
    'children' => $children,
    'total_price' => $calculatedTotalPrice,
    'tenant_paid_amount' => $tenantPaidAmount,
    'host_amount' => $hostAmount,
    'payment_type' => 'full',
    'status' => 'confirmed',
]);
```

Лістинг А.6 – Формування та надсилання повідомлення власнику про нове бронювання

```
private function sendBookingNotificationToOwner(
    Booking $booking,
    Property $property,
    array $priceDetails = []
): void {
    if ($property->user_id === Auth::id()) {
        return;
    }
}
```

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		94

```

}
$tenant = Auth::user();
$night = $priceDetails['nights']
    ?? Carbon::parse($booking->start_date)
        ->diffInDays(Carbon::parse($booking->end_date));
$checkIn = Carbon::parse($booking->start_date)->format('d.m.Y');
$checkOut = Carbon::parse($booking->end_date)->format('d.m.Y');
$guestsText = $booking->adults . ' дор.';
if ($booking->children > 0) {
    $guestsText .= ', ' . $booking->children . ' діт.';
}
$extraGuestsCount = (int) ($priceDetails['extra_guests_count'] ?? 0);
$extraGuestPrice = (float) ($priceDetails['extra_guest_price'] ?? 0);
$extraGuestsTotalPrice = (float) ($priceDetails['extra_guests_total_price']
?? 0);
$extraGuestsText = '';
if ($extraGuestsCount > 0) {
    $extraGuestsText = "\nДодаткові гості: {$extraGuestsCount} × "
        . number_format($extraGuestPrice, 0, '.', ' ')
        . " грн × {$night} ноч. = "
        . number_format($extraGuestsTotalPrice, 0, '.', ' ')
        . " грн";
}
$messageBody = "НОВЕ БРОНЮВАННЯ\n\n"
    . "Орендар: {$tenant->name}\n"
    . "Об'єкт: {$property->title}\n"
    . "Період: {$checkIn} – {$checkOut}\n"
    . "Ночей: {$night}\n"
    . "Гості: {$guestsText}"
    . $extraGuestsText
    . "\nЗагальна сума: "
    . number_format($booking->total_price, 0, '.', ' ')
    . " грн";
$message = Message::create([
    'sender_id' => Auth::id(),
    'receiver_id' => $property->user_id,
    'body' => $messageBody,
    'reply_to_id' => null,
    'is_read' => false,
]);
try {
    broadcast(new MessageSent($message->load('replyTo'))->toOthers());
} catch (\Exception $e) {
    Log::error('Booking notification broadcast error: ' . $e-
->getMessage());
}
}

```

Лістинг А.7 – Скасування бронювання користувачем

```

public function cancel(Booking $booking)
{

```

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		95

```

if ($booking->user_id !== Auth::id()) {
    abort(403, 'У вас немає прав для скасування цього бронювання.');
```

```

}
if ($booking->status === 'cancelled') {
    return redirect()->back()->withErrors([
        'error' => 'Бронювання вже скасовано.',
    ]);
}
if (Carbon::parse($booking->end_date)->startOfDay()->lte(Carbon::today()))
{
    return redirect()->back()->withErrors([
        'error' => 'Завершене бронювання не можна скасувати.',
    ]);
}
$booking->load('property');
$booking->update([
    'status' => 'cancelled',
]);
$this->sendCancellationNotificationToOwner($booking);
return redirect()->back()->with('success', 'Бронювання успішно
скасовано.');
```

```

}
private function sendCancellationNotificationToOwner(Booking $booking): void
{
    $property = $booking->property;
    if (!$property || $property->user_id === Auth::id()) {
        return;
    }
    $guest = Auth::user();
    $checkIn = Carbon::parse($booking->start_date)->format('d.m.Y');
    $checkOut = Carbon::parse($booking->end_date)->format('d.m.Y');
    $messageBody = "БРОНЮВАННЯ СКАСОВАНО ГОСТЕМ\n\n"
        . "Гість: {$guest->name}\n"
        . "Об'єкт: {$property->title}\n"
        . "Період: {$checkIn} – {$checkOut}\n\n"
        . "Дати знову доступні для бронювання.";
    $message = Message::create([
        'sender_id' => Auth::id(),
        'receiver_id' => $property->user_id,
        'body' => $messageBody,
        'reply_to_id' => null,
        'is_read' => false,
    ]);
    try {
        broadcast(new MessageSent($message->load('replyTo'))->toOthers());
    } catch (\Exception $e) {
        Log::error('Booking cancellation notification error: ' . $e-
>getMessage());
    }
}
}

```

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		96

Лістинг А.8 – Створення відгуку після завершення проживання

```
public function store(Request $request, Property $property)
{
    $hasStayed = Booking::where('user_id', Auth::id())
        ->where('property_id', $property->id)
        ->where('status', 'confirmed')
        ->where('end_date', '<', now())
        ->exists();

    if (!$hasStayed) {
        return back()->withErrors([
            'message' => 'Ви можете залишити відгук тільки після фактичного
завершення підтвердженої поїздки.',
        ]);
    }

    $validated = $request->validate([
        'comment' => 'required|string|min:5|max:1000',
        'rating' => 'required|integer|min:1|max:5',
    ]);

    Review::create([
        'user_id' => Auth::id(),
        'property_id' => $property->id,
        'comment' => $validated['comment'],
        'rating' => $validated['rating'],
    ]);

    return back()->with('success', 'Дякуємо! Ваш відгук успішно
опубліковано.');
```

Лістинг А.9 – Додавання або видалення житла з обраного

```
public function toggle(Property $property)
{
    $user = Auth::user();

    $user->favoriteProperties()->toggle($property->id);

    return back()->with('status', 'Список обраного оновлено');
}
```

Лістинг А.10 – Надсилання повідомлення в чаті

```
public function store(Request $request)
{
    $validated = $request->validate([
        'receiver_id' => 'required|exists:users,id',
        'body' => 'required|string',
        'reply_to_id' => 'nullable|exists:messages,id',
    ]);

    $message = Message::create([
        'sender_id' => Auth::id(),
```

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		97

```

        'receiver_id' => $validated['receiver_id'],
        'body' => $validated['body'],
        'reply_to_id' => $validated['reply_to_id'] ?? null,
        'is_read' => false,
    ]);
    try {
        broadcast(new MessageSent($message->load('replyTo'))->toOthers());
    } catch (\Exception $exception) {
        Log::error('Socket error: ' . $exception->getMessage());
    }
    if ($request->wantsJson()) {
        return response()->json([
            'message' => $message,
            'unread_messages_count' => $this->getUnreadMessagesCount(Auth::id()),
        ]);
    }
    return back();
}

```

Лістинг А.11 – Створення оголошення та збереження фотографій

```

public function store(Request $request)
{
    if ($request->user()?->role === 'admin') {
        abort(403, 'Адміністратор не може створювати нові оголошення.');
```

```

    }

    $validated = $request->validate([
        'title' => 'required|string|max:255',
        'description' => 'required|string',
        'type' => 'required|string|max:100',
        'address' => 'required|string|max:255',
        'city' => 'required|string|max:255',
        'price_per_night' => 'required|numeric|min:0',
        'max_guests' => 'required|integer|min:1',
        'base_guests_count' => 'required|integer|min:1|lte:max_guests',
        'extra_guest_price' => 'required|numeric|min:0',
        'latitude' => 'required|numeric|between:-90,90',
        'longitude' => 'required|numeric|between:-180,180',
        'images' => 'required|array|min:3',
        'images.*' => 'required|image|mimes:jpeg,jpg,png|max:2048',
    ], [
        'images.required' => 'Необхідно завантажити фотографії житла.',
        'images.min' => 'Необхідно завантажити щонайменше три фотографії.',
        'images.*.image' => 'Кожен завантажений файл має бути зображенням.',
        'images.*.mimes' => 'Допустимі формати фотографій: JPEG, JPG і PNG.',
        'images.*.max' => 'Розмір кожної фотографії не повинен перевищувати 2
МБ.',
        'base_guests_count.lte' => 'Базова кількість гостей не може
перевищувати максимальну.',
    ]);

    return DB::transaction(function () use ($request, $validated) {

```

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		98

```

$propertyData = collect($validated)
    ->except('images')
    ->toArray();

$property = Property::create(array_merge($propertyData, [
    'user_id' => auth()->id(),
    'is_active' => false,
]));

foreach ($request->file('images') as $index => $image) {
    $path = $image->store('properties', 'public');

    $property->images()->create([
        'path' => $path,
        'is_main' => $index === 0,
    ]);
}

$admin = User::where('role', 'admin')->first();

if ($admin) {
    Message::create([
        'sender_id' => auth()->id(),
        'receiver_id' => $admin->id(),
        'body' => "ЗАЯВКА НА ВЕРИФІКАЦІЮ: #{$property->id} | "
            . "{$property->title} | {$property->city}. "
            . "Ціна: {$property->price_per_night} грн.",
        'is_read' => false,
    ]);
}

return redirect()
    ->route('profile.edit')
    ->with('status', 'Об'єкт надіслано на перевірку!');
});
}

```

Лістинг А.12 – Перевірка кількості фотографій під час редагування оголошення

```

public function update(Request $request, Property $property)
{
    if ($property->user_id !== Auth::id()) {
        abort(403, 'Ви не маєте прав на редагування цього оголошення');
    }

    $validated = $request->validate([
        'title' => [
            'required',
            'string',
            'max:255',
            'regex:/^[p{L}s\-\'\\"«»\""]+$/u',
        ],
        'description' => 'required|string',
        'city' => [

```

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'required',
        'string',
        'regex:/^\[p{L}\s\-\']+\$/u',
    ],
    'price_per_night' => 'required|numeric|min:0',
    'latitude' => 'required|numeric',
    'longitude' => 'required|numeric',
    'deleted_images' => 'nullable|array',
    'deleted_images.*' => 'integer',
    'images.*' => 'image|mimes:jpeg,png,jpg|max:2048',
    'base_guests_count' => 'required|integer|min:1',
    'extra_guest_price' => 'required|numeric|min:0',
]);
$deletedImageIds = collect($request->input('deleted_images', []))
    ->map(fn ($id) => (int) $id)
    ->unique()
    ->values()
    ->all();

$currentImagesCount = $property->images()->count();
$deletedImagesCount = empty($deletedImageIds)
    ? 0
    : $property->images()->whereIn('id', $deletedImageIds)->count();
$newImagesCount = $request->hasFile('images')
    ? count($request->file('images'))
    : 0;

$finalImagesCount = $currentImagesCount - $deletedImagesCount +
    $newImagesCount;

if ($finalImagesCount < 3) {
    return redirect()
        ->back()
        ->withErrors([
            'images' => 'Необхідно залишити або завантажити щонайменше 3
фотографії.',
        ])
        ->withInput();
}

$propertyData = collect($validated)
    ->except(['deleted_images', 'images'])
    ->toArray();

$property->update($propertyData);

if (!empty($deletedImageIds)) {
    $imagesToDelete = $property->images()
        ->whereIn('id', $deletedImageIds)
        ->get();

    foreach ($imagesToDelete as $image) {
        Storage::disk('public')->delete($image->path);
        $image->delete();
    }
}

if ($request->hasFile('images')) {
    foreach ($request->file('images') as $image) {
        $path = $image->store('properties', 'public');
        $property->images()->create(['path' => $path]);
    }
}

```

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				100
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
  }
  return redirect()->back()->with('success', 'Оголошення оновлено');
}

```

Лістинг А.13 – Активація та деактивація оголошення адміністратором

```

$wasActive = $property->is_active;
$isDeactivating = $wasActive
  && $request->has('is_active')
  && !$request->boolean('is_active');
if ($isDeactivating && blank($validated['deactivation_reason'] ?? null)) {
  return back()->withErrors([
    'deactivation_reason' => 'Укажіть причину деактивації оголошення.',
  ]);
}
$propertyData = collect($validated)
  ->except(['images', 'deleted_images', 'deactivation_reason'])
  ->toArray();
$property->update($propertyData);
if (!$wasActive && $property->is_active) {
  if ($property->owner) {
    $property->owner->notify(new PropertyActivated($property));
  }
  Message::create([
    'sender_id' => auth()->id(),
    'receiver_id' => $property->user_id,
    'body' => "Ваше оголошення «{$property->title}» активовано! Тепер воно
доступне для бронювання. Ми також надіслали підтвердження вам на пошту.",
    'is_read' => false,
  ]);
}
if ($wasActive && !$property->is_active) {
  $reason = trim((string) ($validated['deactivation_reason'] ?? ''));
  Message::create([
    'sender_id' => auth()->id(),
    'receiver_id' => $property->user_id,
    'body' => "Оголошення «{$property->title}» деактивовано
адміністратором.\n\n"
    . "Причина: {$reason}\n\n"
    . "Після виправлення повідомте адміністратора в чаті.",
    'is_read' => false,
  ]);
}

```

Лістинг А.14 – Обмеження доступу заблокованого користувача

```

public function handle(Request $request, Closure $next)
{
  if (auth()->check() && auth()->user()->is blocked) {

```

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		101

```

    if ($request->routeIs('logout')) {
        return $next($request);
    }
    if ($request->routeIs('chat.*') || $request->is('chat*')) {
        return $next($request);
    }
    if ($request->is('blocked')) {
        return $next($request);
    }
    return Inertia::render('Errors/Blocked', [
        'message' => 'Ваш акаунт було заблоковано адміністратором. Ви не
можете користуватися сервісом, поки блокування активне.',
    ]);
}
return $next($request);
}

```

		Пилипенко Є.В.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		102