

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

**на тему: Розробка вебзастосунку для управління послугами
відпочинкового комплексу «Luxury Resort»**

Виконала: студентка IV курсу, групи КН-421
спеціальності: 122 Комп'ютерні науки

Таміла БРИЗЦЬКА

Керівник Галина МАРЦІЯШ

Рецензент Михайло ФРИЗ
(ім'я та прізвище)

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

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

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук

_____ Галина МАРЦІЯШ

« 02 » березня 2026 року

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

Бризіцька Таміла Михайлівна

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебзастосунку для управління послугами відпочинкового комплексу «Luxury Resort»

керівник роботи Марціяш Галина Ярославівна

(прізвище, ім'я, по батькові)

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

2. Строк подання студенткою роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мови програмування: Java, TypeScript; фреймворки: Spring Boot; база даних: PostgreSQL; бібліотека React, стандарти IEEE 29148-2018, IEEE 29119, ДСТУ 8302:2015.

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

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу вебзастосунку

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

2 Розробка технічного та робочого проекту

2.1 Розробка структури вебзастосунку

2.2 Створення та верстка сторінок вебзастосунку

2.3 Розробка структури бази даних

2.4 Програмування вебзастосунку

2.4.1 Написання клієнтської частини

2.4.2 Написання серверної частини

2.5 Тестування веб-додатку

3 Спеціальний розділ

3.1 Інструкція з розгортання вебзастосунку

3.2 Інструкція з наповнення вебзастосунку

3.3 Інструкція з популяризації та підтримки вебзастосунку

4 Економічний розділ

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

4.3 Розрахунок витрат на електроенергію

4.4 Розрахунок суми амортизаційних відрахувань

4.5 Обчислення накладних витрат

4.6 Складання кошторису витрат та визначення собівартості веб-додатку

4.7 Розрахунок ціни веб-додатку

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

5 Охорона праці, техніка безпеки та екологічні вимоги

5.1 Кваліфікація виробничих травм

5.2 Дії при виявленні пожежі

6 Висновки

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

5. Перелік графічного матеріалу:

1. Схема структурна програми веб-додатку

2. UML-діаграма варіантів використання

3. ER-діаграма бази даних

4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	20.03.2026
2	Збір і узагальнення інформації	01.05.2026	01.05.2026
3	Написання першого розділу	15.05.2026	15.05.2026
4	Розробка технічного та робочого проєкту	29.05.2026	29.05.2026
5	Написання спеціального розділу	05.06.2026	05.06.2026
6	Розрахунок економічної частини	08.06.2026	08.06.2026
7	Написання розділу охорони праці	09.06.2026	09.06.2026
8	Виконання графічної частини	10.06.2026	10.06.2026
9	Оформлення пояснювальної записки	11.06.2026	11.06.2026
10	Погодження нормоконтролю	12.06.2026	12.06.2026
11	Попередній захист кваліфікаційної роботи	11.06.2026	11.06.2026
12	Захист кваліфікаційної роботи	23.06.2026	23.06.2026

7. Дата видачі завдання: 05 березня 2026 р.

Студент

_____ (підпис)

Таміла БРИЗІЦЬКА

Керівник роботи

_____ (підпис)

Галина МАРЦІЯШ

4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР.....	48
4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи. .	49
4.3. Розрахунок витрат на електроенергію.....	51
4.4. Розрахунок суми амортизаційних відрахувань розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами.....	52
4.5. Обчислення накладних витрат.....	52
4.6. Складання кошторису витрат та визначення собівартості розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами.....	53
4.7. Розрахунок ціни розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами.....	54
4.8. Визначення економічної ефективності і терміну окупності капітальних вкладень.....	54
5 Охорона праці, техніка безпеки та екологічні вимоги.....	56
5.1 Кваліфікація виробничих травм.....	56
5.2 Дії при виявленні пожежі.....	58
Висновки.....	60
Перелік посилань.....	61
Додатки.....	63
Додаток А. Лістинг файлу «pom.xml».....	63
Додаток Б. Лістинг файлу «.env.example».....	70
Додаток В. Лістинг файлу «Room.java».....	71
Додаток Г. Лістинг файлу «Room.java».....	74
Додаток Д. Лістинг файлу «RoomController.java».....	76

АНОТАЦІЯ

Тема кваліфікаційної роботи: Розробка вебзастосунку для управління послугами відпочинкового комплексу «Luxury Resort».

Метою даного проекту є розробка веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами.

Пояснювальна записка складається з 5 розділів.

У загальній частині описуються аналітичний огляд існуючих рішень та аналіз технічного завдання розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами.

У другому розділі технічного завдання представлено процес створення програмного продукту, опис та обґрунтування вибору мов програмування та методу створення веб-додатку, опис бази даних проекту, зовнішнє проектування програми, тестування та налагодження веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами.

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

Розрахунок вартості розробки та економічної ефективності приведено в економічній частині розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами.

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

Обсяг пояснювальної записки 78 сторінок.

До складу кваліфікаційної роботи входить графічна частина, яка складається з структурної схеми програми веб-додатку, UML-діаграми варіантів використання, ER-діаграми бази даних, техніко-економічних показників, що виконані на окремих аркушах формату А1.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		7

ANNOTATION

The qualification thesis presents the development of a web application for the «Luxury Resort» leisure complex with a hotel and additional services. The aim of the work is to automate the processes of room booking, service management and customer record-keeping, and to provide convenient interaction between users and the administration of the complex.

The explanatory note consists of five sections.

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

The second section presents the process of creating the software product: the description and justification of the choice of programming languages and the development method, the description of the project database, the external design of the program, and the testing and debugging of the application.

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

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

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

The explanatory note comprises 78 pages.

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

					<i>2026.KBP.122.421.03.00.00 ПЗ</i>	Арк.
						8
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>		

ВСТУП

У сучасному цифровому суспільстві розвиток інформаційних технологій стрімко охоплює всі сфери людської діяльності, зокрема й сферу туризму та готельно-ресторанного бізнесу. Автоматизація процесів бронювання номерів, управлінням послугами, обліку клієнтів і організації відпочинку стає важливим чинником ефективної діяльності відпочинкових комплексів. В умовах високої конкуренції та зростаючих очікувань клієнтів виникає потреба у створенні зручного, функціонального та надійного веб-додатку, який забезпечить якісну взаємодію між користувачами та адміністрацією комплексу.

Кваліфікаційна робота присвячено розробці веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами. Основною метою розробки є оптимізація процесів бронювання номерів, управління додатковими сервісами (SPA, ресторани, розваги), спрощення доступу клієнтів до інформації та підвищення зручності користування послугами комплексу. Реалізація такого рішення дозволяє покращити якість обслуговування клієнтів, зменшити навантаження на персонал, забезпечити прозорість операції і підвищити ефективність управління бізнесом.

Під час реалізації проєкту сучасні технології: на стороні серверної частини – Java з використанням фреймворку Spring Boot і СУБД PostgreSQL, а на стороні клієнта – мова TypeScript та бібліотека React. Для полегшення розгортання застосовано контейнеризацію за допомогою Docker, що забезпечує простоту масштабування та підтримки.

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

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дат		

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

Для ефективної розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами необхідно провести детальний аналіз конкурентного середовища, зокрема вивчити функціональні можливості, переваги та недоліки провідних гравців на ринку онлайн-платформ з нерухомості. Такий підхід дозволяє виявити актуальні тренди, проблеми, які залишаються невирішеними, та сформувані унікальні конкурентні переваги майбутнього сервісу. Серед найбільш популярних платформ у сфері туризму та бронювання варто виділити Booking.com, Airbnb та TripAdvisor, кожна з яких має свої особливості та цільову аудиторію.

1. Booking.com

Booking.com є однією з найпопулярніших міжнародних платформ для бронювання готелів та інших варіантів проживання. Сервіс орієнтований як на туристів, так і на власників готелів, забезпечуючи широкий функціонал для управління бронюванням.

Переваги:

- великий вибір готелів, апартаментів та інших варіантів проживання по всьому світу;
- зручна система фільтрів: ціна, рейтинг, розташування, послуги, тип житла;
- можливість онлайн-бронювання з миттєвим підтвердженням;
- наявність відгуків користувачів, що підвищує довіру до сервісу.

Недоліки:

- висока конкуренція між готелями, що ускладнює просування нових комплексів;
- комісії за бронювання для власників бізнесу;
- обмежені можливості персоналізації власного бренду готелю.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		10

2. Airbnb

Дана платформа дозволяє орендувати житло безпосередньо у власників, а також пропонує унікальні варіанти проживання та досвід локального туризму.

Переваги:

- широкий вибір унікального житла;
- можливість прямої комунікації між орендодавцем і клієнтом;
- гнучкі умови бронювання та індивідуальний підхід;
- Доступ до великої кількості варіантів проживання в різних країнах усього світу.

Недоліки:

- відсутність стандартизації якості послуг;
- менший акцент на комплексному обслуговуванні;
- можливі ризики недостовірної інформації або скасування бронювання;
- відсутність єдиних стандартів сервісу для всіх об'єктів розміщення.

3. TripAdvisor.

Є популярною платформою для пошуку відгуків про готелі, ресторани та туристичні об'єкти.

Переваги:

- велика кількість відгуків і рейтинг від реальних користувачів;
- можливість порівняння різних варіантів проживання;
- інтеграція з іншими сервісами бронювання.

Недоліки:

- відсутність повноцінної системи бронювання в межах сервісу;
- обмежена функціональність для управління готельним бізнесом.

У цьому контексті веб-додатку для відпочинкового комплексу «Luxury Resort» буде розроблений із врахуванням недоліків існуючих платформ, пропонуючи сучасний інтерфейс, автоматизацію основних операцій.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дат		

1.2 Технічне завдання

Метою створення веб-додатку для відпочинкового комплексу «Luxury Resort» є розробка функціонального, безпечного та масштабованого онлайн-сервісу для взаємодії між відпочинковим комплексом та клієнтами, які бажають забронювати номер, скористатися додатковими послугами або отримати інформацію про доступні варіанти відпочинку. Застосунок повинен об'єднувати у собі переваги сучасного інтерфейсу, гнучкого пошуку, систему онлайн-бронювання та можливості керування послугами комплексу. Сервіс включатиме публічну частину для всіх відвідувачів, авторизовані можливості для зареєстрованих користувачів та панель адміністратора для керування контентом, бронювання й діяльності комплексу.

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

Серед основних функціональних можливостей для кінцевого користувача: реєстрація, авторизація, редагування особистого профілю, перегляд доступних номерів, онлайн бронювання, перегляд інформації про додаткові послуги комплексу (SPA, басейн, ресторан, конференц-зали, розваги), створення списку вибраних номерів, надсилання заявок або повідомлень адміністрації, а також використання розширеної системи пошуку за такими параметрами як тип номера, ціна, кількість місць, наявність додаткових послуг, дата бронювання та категорії відпочинку. Кожен номер або послуга будуть представлене у вигляді окремої

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		12

картки з фотографіями, описом, переліком зручностей та ціною.

Для персоналу та адміністратора комплексу буде реалізовано особистий кабінет із можливістю додавання нових номерів, редагування існуючих, перегляду статистики за переглядами, управління запитом від користувачів та керування контактними даними, а також обробка запитів від клієнтів.

Панель адміністратора надаватиме доступ до всіх бронювань і послуг у системі. Адміністратор зможе підтверджувати або скасовувати бронювання, переглядати аналітику за періодами, популярністю послуг і категоріями номерів. Важливою складовою буде автоматична перевірка правильності введення даних, обмеження доступу до окремих функцій для неавторизованих користувачів і модерація контенту.

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

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

1.2.1 Найменування та область застосування

Кваліфікаційна робота на тему «Розробка веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами» є інформаційно-функціональною платформою, що призначена для автоматизації процесів, пов'язаних з бронюванням номерів, управлінням послугами та організацією

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		13

відпочинку клієнтів. Сервіс «Luxury Resort» покликаний забезпечити сучасне середовище для ефективної взаємодії між трьома основними сторонами: адміністрацією комплексу, персоналом та відвідувачами. Платформа дозволяє користувачам переглядати доступні номери, бронювати проживання, ознайомлюватися з додатковими послугами комплексу та керувати власним бронюванням через особистий кабінет.

Область застосування веб-додатку охоплює як клієнтів, які бажають організувати відпочинок, забронювати номер чи скористатися додатковими послугами комплексу, так і адміністрацію відпочинкового комплексу. Окрему групу користувачів складають адміністратори платформи, які здійснюють модерацию контенту, аналітичний контроль та загальне керування функціонуванням системи.

Функціональність застосунку дозволяє охопити всі етапи організації відпочинку: від перегляду інформації про номери та послуги онлайн-бронювання, керування замовленнями, перегляду історії бронювань і взаємодії з адміністрацією комплексу. Таким чином, «Luxury Resort» виступає не просто інформаційним ресурсом, а повноцінною системою автоматизації діяльності відпочинкового комплексу, що підвищує якість обслуговування, спрощує взаємодію з клієнтами та сприяє цифровій трансформації сфери тризму й готельно-ресторанного бізнесу.

Застосунок побудований за клієнт-серверною архітектурою та функціонує у вигляді веб-додатку, доступного з будь-якого пристрою з браузером. Система не потребує встановлення додаткового програмного забезпечення на пристрої користувача, що значно розширює її доступність і зручність використання. Усі дані зберігаються у захищеній базі даних, доступ до якої контролюється відповідно до ролей користувачів – гість, адміністратор, менеджер, рецепсionіст. Діаграму варіантів використання системи, що відображає взаємодію гостя, працівника комплексу та адміністратора з функціоналом застосунку, наведено в графічній частині кваліфікаційної роботи. Це дозволяє забезпечити високий рівень безпеки, персоналізації та гнучкого керування ресурсами платформи.

Окрему увагу в застосунку приділено UX/UI дизайну, який адаптований до

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		14

потреб різних категорій користувачів, включаючи людей з базовим рівнем цифрової грамотності. Інтерфейс «Luxury Resort» реалізований таким чином, щоб мінімізувати кількість дій для досягнення цілей користувача.

У результаті застосунок «Luxury Resort» позиціонується як універсальний веб-додаток, що поєднує інтереси як клієнтів, так і бізнес-сегменту, підвищуючи ефективність, управління, якість сервісу та рівень цифровізації туристичної галузі. Завдяки поєднанню технологічної надійності, сучасного дизайну та багатофункціональності, він може успішно впроваджуватись у якості комерційного продукту, інфраструктурного рішення або науково-освітнього кейсу.

1.2.2 Призначення розробки

Призначенням розробки веб-додатку «Luxury Resort» є автоматизація діяльності відпочинкового комплексу шляхом створення сучасного цифрового інструменту, який забезпечить ефективне управління номерним фондом, бронюванням, додатковими послугами та взаємодією між адміністрацією комплексу й клієнтами. У сучасних умовах стрімкого розвитку технологій, а також зростання вимог клієнтів до швидкості й зручності отримання послуг, особливо актуальним стає впровадження інноваційних програмних рішень у сфері туризму та готельно-ресторанного бізнесу. Веб-додаток «Luxury Resort» покликаний не лише замінити застарілі паперові або фрагментовані електронні засоби обліку та бронювання, а й створити єдину екосистему для організації, комфортного відпочинку та ефективного управління комплексом.

Одна з головних функцій веб-додатку – облік номерів і послуг відпочинкового комплексу. Система передбачає створення єдиної бази даних, у якій кожен номер матиме детальний опис категорію номера, кількість місць, перелік зручностей, вартість проживання, фотографії, статус доступності та додаткові характеристики. Окремо в системі зберігається інформація про додаткові послуги. Це дозволить адміністрації оперативна оновлювати інформації, контролювати актуальність даних та забезпечувати високий рівень обслуговування

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		15

клієнтів.

Застосунок забезпечує зручний і гнучкий інструмент пошуку та фільтрації номерів за основними параметрами, такими як вартість, категорію номера, кількість місць та іншими характеристиками. Це дозволяє клієнтам швидко знайти найбільш релевантні варіанти, що відповідають їхнім індивідуальним критеріям. Завдяки цьому, користувачі отримують зручний інтерфейс для вибору оптимального варіанту для відпочинку.

Додатковою перевагою є можливість бронювання номерів та замовлення додаткових послуг. Користувачі можуть в один клік оформити бронювання або замовити окремі послуги. Усі запити фіксуються в системі та мають статус (наприклад, "в обробці", "підтверджено", "відхилено"), що дозволяє прозоро відслідковувати історію взаємодії. Для адміністраторів реалізовано сповіщення про нові заявки з можливістю перегляду деталей і швидкої відповіді клієнту, що пришвидшує процес комунікації.

Кабінет користувача є важливою складовою персоналізації сервісу. Залежно від ролі, користувач отримує різний набір функціоналу. Для гостей передбачено можливість збереження об'єктів до "вибраного" та керування особистими даними. Для менеджерів доступні інструменти додавання об'єктів, перегляду заявок, оновлення статусів, створення статистичних звітів. Адміністратори системи мають розширений доступ для керування користувачами, контентом і аналітикою.

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

Розробка вебзастосунку «Luxury Resort» орієнтована на сучасні технології. Серверна частина реалізується з використанням Java та фреймворку Spring Boot, що забезпечує надійність, безпеку та масштабованість. У якості бази даних

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		16

використовується PostgreSQL — потужна реляційна СУБД з підтримкою геоданих (PostGIS), що є критично важливим для роботи з мапами. Фронтенд частина створюється на основі React з TypeScript, що дозволяє розробити інтуїтивно зрозумілий інтерфейс, гнучкий до змін.

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

Розробка вебзастосунку «Luxury Resort» спрямована на підвищення ефективності відпочинкового комплексу шляхом автоматизації його діяльності, зниження навантаження на персонал, покращення комунікації з клієнтами та формування позитивного іміджу інноваційного комплексу. У поєднанні з сучасними вебтехнологіями та фокусом на зручність користувача, додаток є перспективним рішенням для впровадження у практичну діяльність ринку готельного бізнесу України.

1.2.3 Вимоги до функціоналу вебзастосунку

Для забезпечення ефективної роботи веб-додаток «Luxury Resort» повинен включати таку функціональність:

- збереження інформації про номери та послуги комплексу з детальними характеристиками;
- організація даних за категоріями, цінами, статусами тощо;
- пошук та фільтрація об'єктів за заданими критеріями (наприклад, ціна, кількість місць, категорія номера);
- можливість гостям здійснювати онлайн-бронювання номерів і замовлення додаткових послуг;
- модуль управління доступом із можливістю створення профілів для адміністраторів, менеджерів і гостей;
- панель аналітики для відстеження статистики запитів і популярності номерів.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		17

1.2.4 Вимоги до програмної документації

По закінченню розробки програмного забезпечення потрібно підготувати таку документацію:

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

1.2.5 Техніко-економічні показники

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

- собівартість розробки – 72 151 грн;
- планова ціна продукту – 112 556 грн;
- плановий прибуток від реалізації – 40 405 грн;
- чиста теперішня вартість (ЧТВ) – 28 330 грн.; термін окупності капітальних вкладень – 2,1 року.

1.2.6 Стадії та етапи розробки

Розробка веб-додатку «Luxury Resort» буде мати такі стадії:

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

Кожна зі стадій є логічним продовженням попередньої та завершується проміжним результатом, який використовується на наступному етапі. Такий поетапний підхід дозволяє контролювати хід виконання робіт, своєчасно виявляти

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дат		

й усунути недоліки та забезпечити високу якість кінцевого програмного продукту.

1.2.7 Порядок тестування та прийому

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

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

Кожен етап тестування супроводжується фіксацією виявлених недоліків та їх подальшим усуненням. Функціональне тестування передбачає перевірку коректності роботи основних сценаріїв: реєстрації та авторизації користувачів, бронювання номерів, замовлення додаткових послуг і формування звітів. Окремо проводиться перевірка розмежування прав доступу для різних ролей користувачів, а також стійкості системи до некоректного введення даних. Приймання вебзастосунку здійснюється після успішного проходження всіх перевірок та підтвердження відповідності реалізованого функціоналу вимогам технічного завдання.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дат		

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури вебзастосунку

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

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

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

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		20

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

Значну увагу приділено розробці бази даних, яка є основою для зберігання всієї інформації, пов'язаної з номерами, користувачами, заявками, повідомленнями, бронюваннями та історією змін. У якості системи управління базами даних обрано PostgreSQL – надійну реляційну СУБД. Структура бази даних проектується з урахуванням нормалізації, цілісності даних та забезпечення швидкої роботи при масштабуванні. Ключові таблиці включають: User, Booking, Room, Payment, тощо. Використання зовнішніх ключів забезпечує зв'язки між таблицями, дозволяючи зберігати цілісність інформації.

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

Таким чином, структура веб-додатку «Luxury Resort» побудована за принципами сучасної веб-розробки: розподілена архітектура, розмежування обов'язків, використання перевірених технологій, захищений обмін даними між клієнтом і сервером.

Це дозволяє створити стабільний, надійний і зручний інструмент для автоматизації відпочинкового комплексу, який легко розширюється й адаптується під потреби кінцевого користувача. Структурну схему програми наведено в графічній частині кваліфікаційної роботи.

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		21

2.2 Створення та верстка сторінок вебзастосунку

Процес створення та верстки сторінок веб-додатку «Luxury Resort» є важливим етапом реалізації клієнтської частини системи, оскільки саме інтерфейс взаємодії з користувачем формує перше враження про сервіс і визначає зручність користування. Основною метою на цьому етапі є розробка зручного, інтуїтивно зрозумілого дизайну, що враховує сучасні тенденції у веброзробці, потреби користувачів та специфіку туристичного ринку. Створення інтерфейсу здійснювалося з використанням бібліотеки React у поєднанні з мовою програмування TypeScript, яка дозволяє швидко реалізувати стилізовані компоненти з дотриманням принципів адаптивного дизайну.

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

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

На сторінці детального перегляду номеру представлена розширена

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		22

інформація: фотографії, опис, характеристики,. З технічної точки зору ця сторінка динамічно підвантажує інформацію з бази даних відповідно до ідентифікатора обраного об'єкта, що передається через параметри URL.

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

Також була реалізована сторінка додавання нового номеру, яка доступна лише адміністраторам. Ця форма включає обов'язкові поля: назва, опис, ціна, тип,, кількість кімнат, фото. Для завантаження зображень реалізовано попередній перегляд (preview) з можливістю видалення до підтвердження збереження. Всі поля проходять перевірку на коректність введення даних, а при відправленні форми інформація зберігається в базу даних PostgreSQL.

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

Таким чином, створення та верстка сторінок веб-додатку «Luxury Resort» реалізовано відповідно до сучасних принципів розробки вебінтерфейсів. Результатом є інтуїтивно зрозумілий, динамічний і адаптивний інтерфейс, який задовольняє потреби як пересічних користувачів, так і працівників комплексу. Це забезпечує високий рівень зручності, функціональності та довіри до сервісу.

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		23

2.3 Розробка структури бази даних

Для зберігання та обробки інформації у веб-додатку «Luxury Resort» розроблена реляційна структура бази даних, яка забезпечує ефективне управління номерами, послугами, користувачами, бронюванням та звітністю. Взаємозв'язки між таблицями реалізовано за допомогою зовнішніх ключів, що гарантує цілісність даних. Основу СКБД становить PostgreSQL. Нижче наведено детальний опис основних таблиць бази даних. Повну ER-діаграму бази даних наведено в графічній частині кваліфікаційної роботи.

1) Таблиця «rooms» містить основну інформацію про номери, що розміщені на платформі:

- id,
- name
- description
- base_price_per_night,
- room_type,
- max_occupancy,
- size_sqm,
- floor,
- room_number,
- status,
- amenities,
- images,
- avg_rating,
- review_count,
- created_at,
- updated_at.
-

2) Таблиця «services» містить інформацію про доступні послуги в комплексі

- id,

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		24

- name,
- description
- category,
- price,
- duration_minutes,
- max_participants,
- images,
- is_available,
- popularity_score,
- created_at

3) Таблиця «reviews» зберігає відгуки про номери або послуги представлені у комплексі:

- id,
- room_id,
- user_id,
- booking_id,
- rating,
- comment,
- images,
- sentiment_score,
- is_approved,
- created_at

4) Таблиця «users» зберігає інформацію про користувачів вебзастосунку (як клієнтів, так і працівників):

- first_name,
- last_name,
- email, phone,
- password_hash,
- role,
- is_active,

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		25

- loyalty_points,
- created_at,
- updated_at.

5) Таблиця «booking» зберігає інформацію про заявки на бронювання:

- id,
- room_id,
- user_id,
- check_in_date,
- check_out_date,
- guests_count,
- base_total,
- dynamic_price_total,
- final_multiplier,
- pricing_snapshot,
- status,
- cancellation_reason,
- special_requests,
- loyalty_points_earned,
- loyalty_points_used,
- created_at,
- updated_at

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

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		26

2.4 Програмування вебзастосунку

Процес програмування веб-додатку «Luxury Resort» є одним з ключових етапів у реалізації функціональності системи управління відпочинкового комплексу. Основна мета програмування – реалізувати усі логічні компоненти, інтегрувати базу даних, реалізувати зручний і сучасний інтерфейс користувача, а також забезпечити надійну комунікацію між клієнтською та серверною частинами. Для досягнення поставлених цілей було обрано стек технологій, який поєднує Java, фреймворк Spring Boot на бекенді, PostgreSQL як систему керування базами даних, та TypeScript з бібліотекою React для фронтенд-розробки.

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

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

Бекенд також включає реалізацію API для обробки записів перегляду номерів, створення заявок бронювання, ведення звітів і переглядами тощо. Запити реалізовані згідно з REST-принципами, використовуються HTTP-методи GET,

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		27

POST, PUT, DELETE, що відповідають відповідно операціям читання, створення, оновлення та видалення ресурсів. При обробці запитів використовується валідація даних через анотації Hibernate Validator (наприклад, @NotNull, @Size, @Email), що дозволяє зменшити кількість помилок і забезпечити коректність вхідних даних.

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

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

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

Окремо реалізовані інтерфейси кабінетів для різних типів користувачів. Наприклад, клієнт бачить свої заявки та доступні послуги, може залишити заявку на перегляд. Адміністратор має повний контроль над всіма розділами сайту, включаючи послуги, номерів, користувачів та модерацію контенту.

Розробка велась у локальному середовищі з використанням системи

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		28

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

Таким чином, програмування веб-додатку «Luxury Resort» включало повний цикл реалізації логіки на бекенді та фронтенді, з урахуванням сучасних стандартів безпеки, масштабованості, зручності використання та модульності.

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

2.4.1 Написання клієнтської частини

Клієнтська частина веб-додатку «Luxury Resort» є важливою складовою всієї системи, оскільки саме через неї здійснюється безпосередня взаємодія користувача з функціоналом сервісу. Основна мета клієнтської частини полягає у створенні зручного, інтуїтивно зрозумілого, швидкого й адаптивного інтерфейсу, що дозволяє користувачам переглядати доступні послуги, фільтрувати пропозиції, реєструватися, авторизовуватися, здійснювати бронювання, тощо.

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

Проект було реалізовано з використанням компонентно-орієнтованого підходу: кожен елемент інтерфейсу – форма, список об'єктів, фільтр, сторінка перегляду тощо – представлений окремим функціональним компонентом. Для керування станом застосунку використано React Hooks (useState, useEffect,

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		29

useContext) у поєднанні з контекстним API, що дозволяє централізовано зберігати дані про авторизованого користувача, обрані об'єкти, активні фільтри та інші глобальні стани.

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

Після запуску веб-додатку «Luxury Resort» завантажує головну сторінку зі списком актуальних номерів та послуг. Основний інтерфейс складається з фільтрів (за типом, ціною, рейтингом, кількістю кімнат), номерів, пагінації та пошукового поля. Кожна картка включає ключову інформацію – фото, назву, ціну, статус – а також кнопки для бронювання або переходу на детальну сторінку.

Окрема увага приділялася розробці сторінки перегляду конкретного номеру. На ній виводиться повна інформація про номер, а також можливість залишити заявку бронювання.

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

Користувач має можливість переглянути свої заявки та перегляди у власному кабінеті (ProfilePage). Якщо користувач є адміністратором, йому доступна можливість керувати послугами та номерами (додавати, редагувати, видаляти), переглядати звіти та змінювати статуси запитів.

Візуальне оформлення клієнтської частини реалізовано за допомогою бібліотек, що забезпечує адаптивний, сучасний та уніфікований інтерфейс із використанням компонентів, таких як Button, Card, TextField, DataGrid, Dialog, AppBar, Tabs, тощо. Сторінки стилізовані з урахуванням принципів доступності та зрозумілості.

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		30

Маршрутизація у клієнтському застосунку організована за допомогою react-router-dom. Всі основні сторінки, такі як /, /login, /register, /apartments, /favorites, /profile, /admin, мають окремі компоненти, що відповідають за логіку та візуальне представлення. У випадку переходу на неіснуючий маршрут відображається сторінка 404.

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

Таким чином, клієнтська частина вебзастосунку реалізує всі основні функції взаємодії з користувачем та забезпечує логіку для роботи із веб-додатком «Luxury Resort». Її реалізація ґрунтується на сучасних принципах компонентного підходу, реактивності, повторного використання логіки та модульності. В результаті створено надійний, масштабований і зручний фронтенд, що дозволяє якісно та ефективно вирішувати задачі користувачів відпочинкового комплексу.

2.4.2 Написання серверної частини

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

Для побудови серверної частини було обрано стек технологій на основі Java 17 та Spring Boot. Цей фреймворк дозволяє створювати RESTful-сервіси з чіткою структурою, високою продуктивністю та гнучкими можливостями конфігурації. Доступ до бази даних здійснюється за допомогою Spring Data JPA, що забезпечує

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		31

прозору інтеграцію з PostgreSQL.

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

а) Сутності (entity):

- AuditLog;
- Booking;
- Payment;
- RefreshToken;
- Review;
- Room.
- ServiceEntity
- ServiceOrder
- User

б) Репозиторії (repository):

- AuditLogRepository;
- BookingRepository;
- PaymentRepository;
- RefreshTokenRepository;
- ReviewRepository;
- RoomRepository.
- ServiceEntityRepository
- ServiceOrderRepository
- UserRepository.

Рівень доступу до даних у вебсервері реалізується за допомогою репозиторіїв та сервісів. Використано Spring Data JPA, який дозволяє легко взаємодіяти з базою даних за допомогою інтерфейсів та анотацій. Ці сервіси та репозиторії допомагають взаємодіяти з базою даних, ізолюючи бізнес-логіку від прямих взаємодій з джерелом даних. В проекті наведені наступні репозиторії:

- AuditLogRepository;
- BookingRepository;

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		32

- PaymentRepository;
- RefreshTokenRepository;
- ReviewRepository;
- RoomRepository.
- ServiceEntityRepository
- ServiceOrderRepository
- UserRepository.

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

Нижче наведено опис основних ендпоінтів.

- 1) Взаємодія із обліковими записами та токенами:
 - POST `/auth/register`, `/auth/login`, `/auth/refresh`.
- 2) Отримання номерів:
 - GET `/rooms`, `/rooms/{id}`.
- 3) Управління номерами:
 - POST/PUT/DELETE `/rooms`.
- 4) Робота з бронюваннями:
 - POST/GET `/bookings`, GET `/bookings/{id}`.
- 5) Робота із статусом бронювання з клієнтської сторони:
 - PUT `/bookings/{id}/status`.
- 6) Отримання інвойсу про бронювання:
 - GET `/bookings/{id}/invoice`.
- 7) Оплата бронювання:
 - POST `/bookings/{id}/pay`.
- 8) Каталог послуг:
 - GET `/services`, `/services/{id}`.
- 9) Популярні послуги:
 - GET `/services/recommendations`.
- 10) Операції з послугами:

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дат		

- POST/PUT/DELETE `/services`.
- 11) Замовлення послуг:
 - POST/GET `/service-orders`, PUT `.../status`.
- 12) Отримання відгуків:
 - GET `/reviews`.14)
- 14) Модерація відгуку:
 - PUT `/reviews/{id}/approve`.
- 15) Видалення відгуку:
 - DELETE `/reviews/{id}`.
- 16) Отримання інформації по програмі лояльності:
 - GET `/loyalty/balance`.
- 17) Отримання інформації по KPI:
 - GET `/admin/analytics/dashboard`.
- 18) Рейтинг номеру:
 - GET `/admin/analytics/room-sentiment`.
- 19) Прогноз зайнятості за 14 дні:
 - GET `/admin/analytics/occupancy-forecast`.
- 20) Користувачі-адміністратори:
 - GET `/admin/users`, PUT `/admin/users/{id}/role`.

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

2.5 Тестування веб-додатку

Переходимо на головну сторінку вебзастосунку. Результат представлено на

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		34

рисунку 2.1.

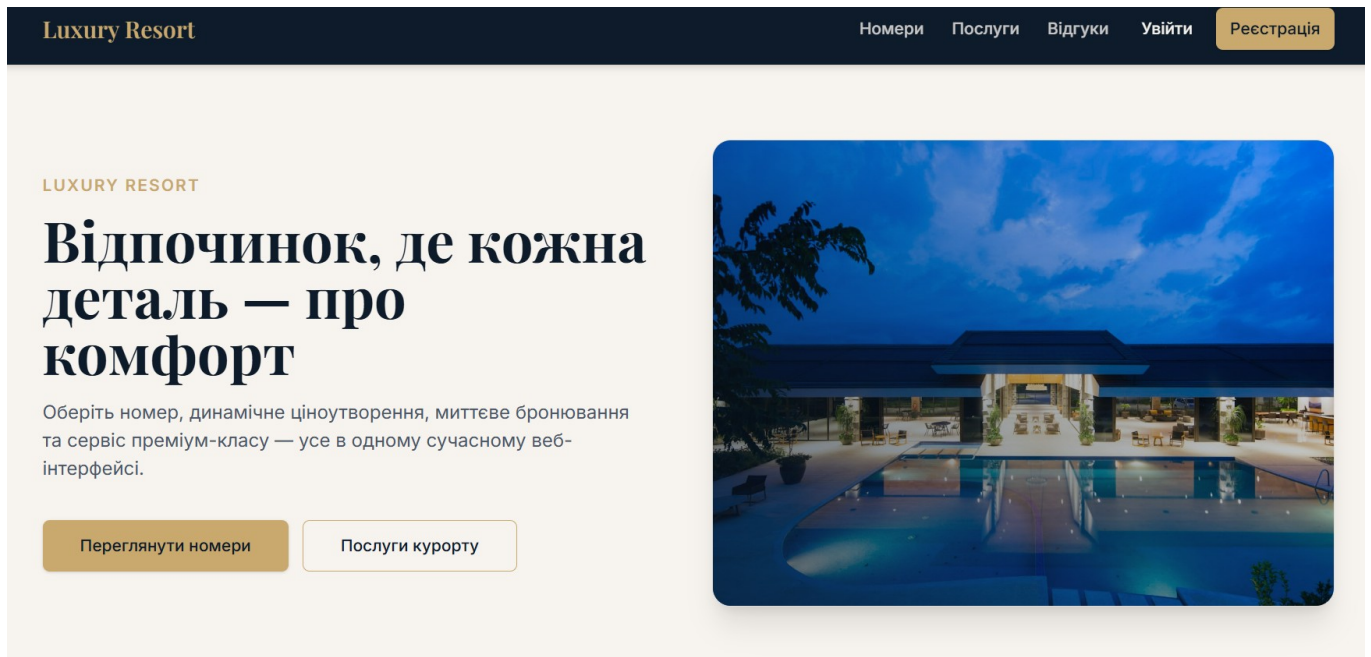


Рисунок 2.1 – Головна сторінка вебзастосунку

Перейдемо в секцію “Відгуки”, щоб впевнитися в надійності комплексу. Результат представлено на рисунку 2.2.

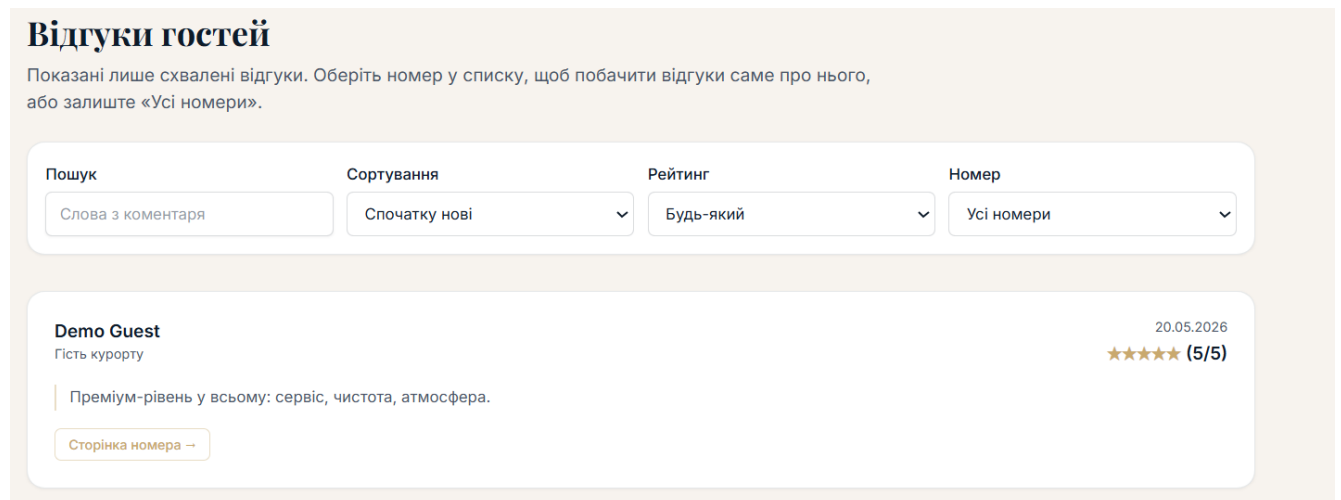


Рисунок 2.2 – Секція «Відгуки»

Впевнившись в надійності перейдемо в секцію “Номери”, можливо там знайдемо потрібний нам номер. Результат представлено на рисунку 2.3.

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		35

Номери

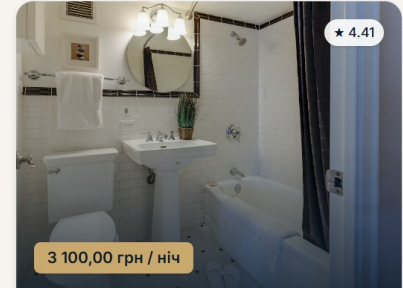
Оберіть категорію та дати бронювання у картці номера.



3 400,00 грн / ніч



4 800,00 грн / ніч



3 100,00 грн / ніч

Рисунок 2.3 – секція «Номери»

Виберемо другий номер. Натискаємо на нього та переходимо на сторінку детального опису. Результат представлено на рисунку 2.4.

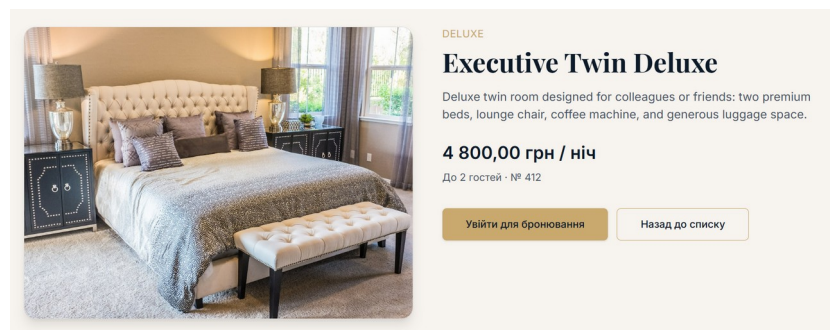


Рисунок 2.4 – Детальна сторінка номеру

Для того, щоб відправити запит бронювання, потрібно створити особистий кабінет. Зареєструємося. Результат представлено на рисунку 2.5.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		36

Реєстрація'."/>

Рисунок 2.5 – Реєстрація

Після чого починаємо оформлення бронювання. Вводимо час бронювання . Результат представлено на рисунку 2.6.

Рисунок 2.6 – Сторінка бронювання

Вводимо кількість людей . Результат представлено на рисунку 2.7.

Рисунок 2.7 – Сторінка бронювання

Переходимо до оплати . Результат представлено на рисунку 2.8.

1. Дати 2. Деталі 3. Оплата 4. Готово

Розбір ціни

Базова сума 158 400,00 грн
 Комбінований множник 1.033984x
 Знижка лояльності 0

До сплати 163 783,07 грн

ЗАСТОСОВАНІ ПРАВИЛА
 Last Minute Premium Demand 1.12x Seven Night Loyalty Stay 0.88x
 Summer Weekend Surge 1.0490909086x
 Нараховано баніле: 1637

СПОСІБ ОПЛАТИ

Картка
 Visa / Mastercard (демо)

Apple Pay
 Шведка оплата

Google Pay
 Android / Google Wallet

Переказ
 Реквізити після підтвердження

Назад [Перейти до оплати](#)

Рисунок 2.8 – Сторінка оплати

Отримуємо повідомлення про успішне бронювання . Результат представлено на рисунку 2.9.

1. Дати 2. Деталі 3. Оплата 4. Готово

Бронювання підтверджено
 Дякуємо за вибір Luxury Resort.

Квитанцію у форматі PDF (оформлена як чек) можна завантажити в «Мої бронювання» — кнопка поруч із бронюванням.

[Мої бронювання](#)

Рисунок 2.9 – Підтвердження бронювання

Переходимо у власний кабінет та переглядаю свої бронювання . Результат представлено на рисунку 2.10.

Мої бронювання

Пошук за номером/назвою

Executive Twin Deluxe · 412
 03.05.2026 — 05.06.2026
 Статус: CONFIRMED
 163 783,07 грн

[Чек PDF](#) [Скасувати](#)

Рисунок 2.10 – Особистий кабінет з бронюванням

Завантажуємо чек про оплату у форматі PDF. Результат представлено на

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		38

рисунку 2.11.

LUXURY RESORT	
КВИТАНЦЯ ПРО ОПЛАТУ БРОНЮВАННЯ	
№ БР/БРОН/КА/105 - 31.05.2026 22:20	
Послужба	Сума
Номер	Executive Twin Deluxe № 412
Гість	Олівер Март oliver.mart@luxuryresort.local
Залишок вноски	2026-05-03 2026-06-05
Номер	33
Базова вартість (33 н.)	4 800,00 € + 33
Підручки бази	158 400,00 €
Додаткові централізовані (+1.034)	163 783,07 €
Статус бронювання	CONFIRMED
Спосіб оплати	Банківська картка
ВВН/транзакція	ТОВ ЗАКРИТЕ
ДО СПЛАТИ	163 783,07 €

Рисунок 2.11 – Чек оплати бронювання

Переходимо в секцію “Послуги” та замовимо одну із них. Результат представлено на рисунку 2.12.

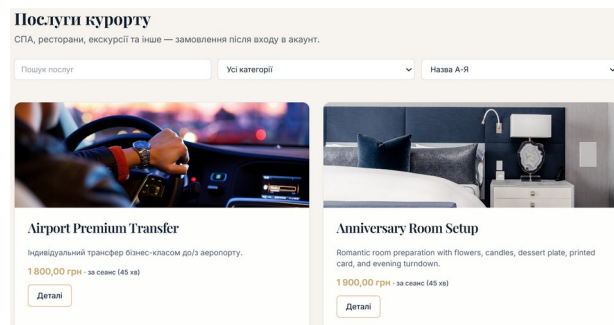


Рисунок 2.12 – Секція «Послуги»

Вибираємо іншу та переходжу на детальну сторінку. Результат представлено на рисунку 2.13.

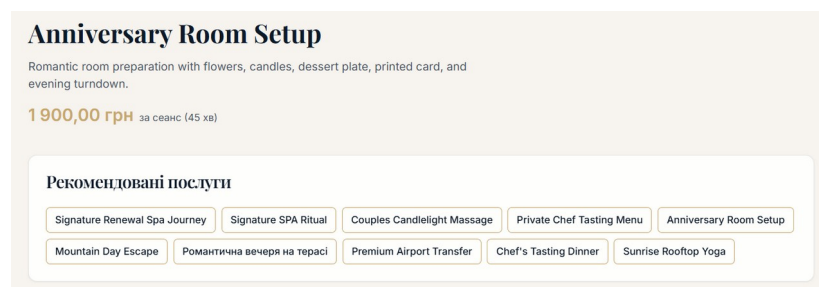


Рисунок 2.13 – детальна сторінка послуги

Починаємо оформлення послуги. Результат представлено на рисунку 2.14.

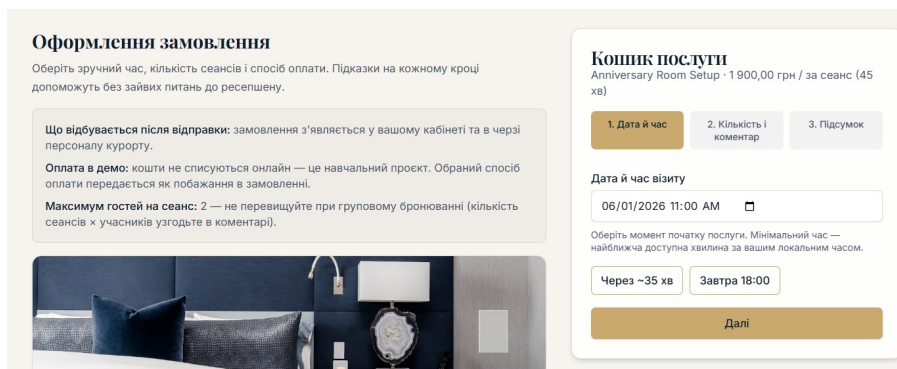


Рисунок 2.14 – Детальна сторінка послуги

Після вводу отримую результат про успішне прийняття замовлення. Результат представлено на рисунку 2.15.

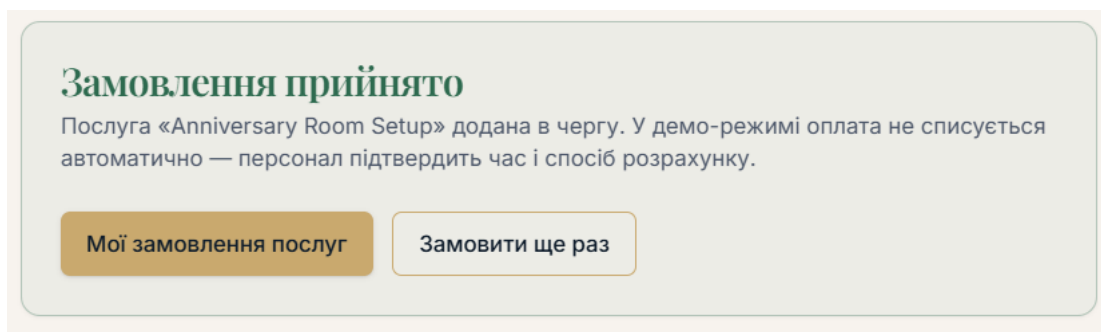


Рисунок 2.15 – Успішно прийняте замовлення

Переходимо в особистий кабінет та переглядаю замовлення. Результат представлено на рисунку 2.16.

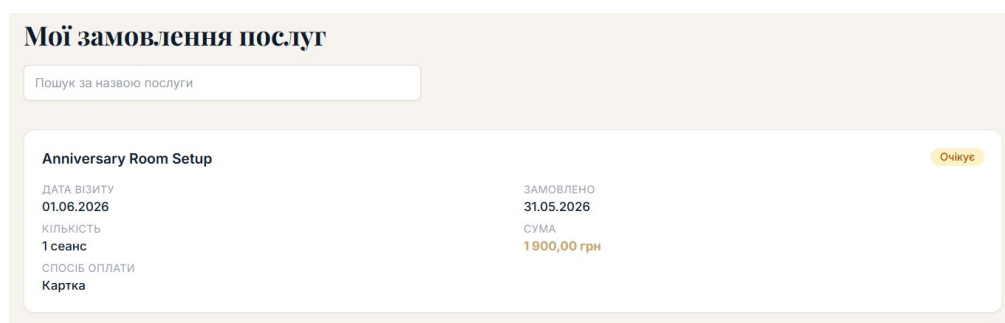


Рисунок 2.16 – Успішно прийняте замовлення

Змінюємо роль на адміністратора та переходжу в його кабінет. Результат

представлено на рисунку 2.17.

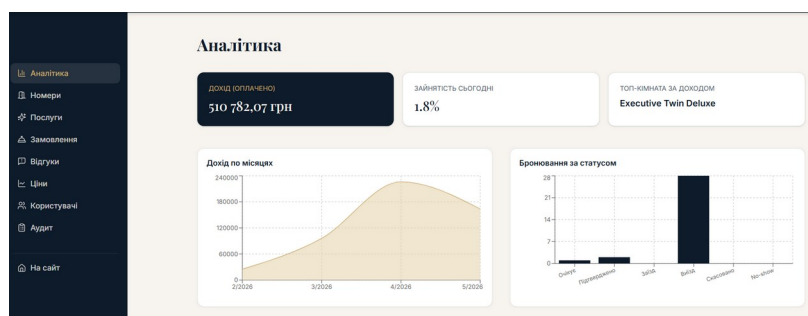


Рисунок 2.17 – Кабінет адміністратора

Тут представлена можливість керування номерами . Результат представлено на рисунку 2.18.

Керування номерами Додати номер

Пошук: назва/№ Новіші

Номер	Назва	Тип	Ціна / ніч	Статус			
281	Azure Standard Room	STANDARD	3 400,00 грн	AVAILABLE	На сайті	Редагувати	Видалити
412	Executive Twin Deluxe	DELUXE	4 800,00 грн	AVAILABLE	На сайті	Редагувати	Видалити
226	Business Compact Room	STANDARD	3 100,00 грн	AVAILABLE	На сайті	Редагувати	Видалити
238	Accessible Deluxe Room	DELUXE	5 200,00 грн	AVAILABLE	На сайті	Редагувати	Видалити
1882	Presidential Residence	PRESIDENTIAL	22 000,00 грн	AVAILABLE	На сайті	Редагувати	Видалити
385	Family Garden Suite	SUITE	8 200,00 грн	AVAILABLE	На сайті	Редагувати	Видалити

Рисунок 2.18 – Керування номерами

Також послугами. Результат представлено на рисунку 2.19.

Каталог послуг (адмін) Додати послугу

Пошук послуг Назва А-Я

Назва	Категорія	Ціна	Активна			
Late Night City Shuttle	TRANSFER	950,00 грн	так	На сайті	Редагувати	Видалити
Mountain Day Escape	EXCURSION	4 200,00 грн	так	На сайті	Редагувати	Видалити
Old Town Private Walk	EXCURSION	1 700,00 грн	так	На сайті	Редагувати	Видалити
Personal Training Session	FITNESS	1 400,00 грн	так	На сайті	Редагувати	Видалити
Premium Airport Transfer	TRANSFER	2 100,00 грн	так	На сайті	Редагувати	Видалити
Private Chef Tasting Menu	DINING	3 800,00 грн	так	На сайті	Редагувати	Видалити

Рисунок 2.19 – Керування послугами

Адміністратор може приймати замовлення та керувати ними, наприклад

змінювати їхній статус. Результат представлено на рисунку 2.20.

Черга замовлень послуг
Оновлення статусів для персоналу готелю.

Пошук: послуга або email Усі статуси Новіші

Послуга	Клієнт	Час	Сума	Статус	
Anniversary Room Setup	аааааааа-аааа-аааа-аааа-аааааааа0101	01.06.2026	1 900,00 грн	PENDING	Зберегти
Premium Airport Transfer	аааааааа-аааа-аааа-аааа-аааааааа0106	05.03.2026	2 100,00 грн	COMPLETED	Зберегти
Private Chef Tasting Menu	аааааааа-аааа-аааа-аааа-аааааааа0107	15.03.2026	7 600,00 грн	COMPLETED	Зберегти
Sunrise Rooftop Yoga	аааааааа-аааа-аааа-аааа-аааааааа0108	20.03.2026	900,00 грн	COMPLETED	Зберегти
Kids Creative Club	аааааааа-аааа-аааа-аааа-аааааааа0104	28.03.2026	1 500,00 грн	COMPLETED	Зберегти
Breakfast in Bed	аааааааа-аааа-аааа-аааа-аааааааа0105	27.02.2026	2 500,00 грн	COMPLETED	Зберегти

Рисунок 2.20 – Керування послугами

А також змінювати та переглядати користувачів. Результат представлено на рисунку 2.21.

Користувачі

Пошук: email/ім'я Усі ролі

Email	Ім'я	Роль	
mark.taylor@example.test	Mark Taylor	GUEST	Зберегти
guest.showcase@luxuryresort.local	Sofia Melnyk	GUEST	Зберегти
emma.stone@example.test	Emma Stone	GUEST	Зберегти
andrii.shevchenko@example.test	Andrii Shevchenko	GUEST	Зберегти
nina.parker@example.test	Nina Parker	GUEST	Зберегти
admin.showcase@luxuryresort.local	Olivia Hart	ADMIN	Зберегти

Рисунок 2.21 – Керування користувачами

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розгортання вебзастосунку

Розгортання веб-додатку «Luxury Resort» передбачає не лише локальне використання, але й надання доступу до сервісу з мережі Інтернет. Для цього потрібно забезпечити наявність білої (публічної) IP-адреси, яка дозволить користувачам підключатись до сервера з будь-якої точки світу. У випадку розгортання на віртуальному приватному сервері (VPS) або хмарному хостингу, така адреса зазвичай надається автоматично. Важливо також відкрити відповідні порти у фаєрволі чи маршрутизаторі: 80 і 443 – для HTTP/HTTPS-доступу до інтерфейсу користувача (frontend), 8080 – для API-запитів до бекенду (якщо вони не проксіювані), порт 5432 бази даних PostgreSQL відкривати не рекомендується з міркувань безпеки, замість цього бажано використовувати VPN або тунелювання.

Після складання клієнтської частини вебзастосунку за допомогою команди `npm run build`, необхідно налаштувати вебсервер, який оброблятиме запити користувачів і віддаватиме статичні ресурси. Найбільш доцільним вибором для цього є Nginx – високопродуктивний вебсервер, який легко інтегрується з Docker та забезпечує проксійний доступ до бекенду. У типовій конфігурації вебсервер слухає порт 80 та обробляє запити до кореневого шляху /, повертаючи файл `index.html`, а запити до `/api/` перенаправляє на Spring Boot-бекенд, що працює на порту 8080. Це дозволяє приховати деталі реалізації серверної частини та централізовано обробляти трафік.

Щоб веб-додаток був доступним за доменним іменем (наприклад, `luxuryresort.com`), необхідно зареєструвати домен та додати до нього DNS-запис типу A, що вказуватиме на публічну IP-адресу сервера. Для забезпечення захищеного з'єднання обов'язково потрібно налаштувати SSL-сертифікат. Найпростіший і безкоштовний спосіб зробити це – скористатись сервісом Let's Encrypt.

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		43

3.2 Інструкція з наповнення вебзастосунку

Ефективне функціонування веб-додатку «Luxury Resort» безпосередньо залежить від правильного і систематичного наповнення системи актуальними, структурованими та достовірними даними. Інструкція з наповнення є ключовим компонентом у впровадженні програмного продукту, оскільки вона визначає алгоритм додавання номерів, послугів, наповнення інформаційних блоків, управління користувацьким контентом, фотоматеріалами, а також формування супровідної інформації, необхідної для коректного функціонування платформи.

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

У першу чергу заповнюється загальна інформація про номер. До цієї категорії входять такі обов'язкові поля: тип номеру, кількість людей, які можуть бути в ньому, вартість, рейтинг. Варто звернути увагу на правильність написання усієї інформації, адже саме за нею буде проводитися пошук конкретних номерів..

Далі слідує секція опису номеру, де необхідно надати детальну текстову інформацію про всі основні деталі номеру, без помилок, із акцентом на ключових перевагах конкретної пропозиції. Це значно підвищує зацікавленість користувача, рівень довіри до платформи.

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

Після завершення заповнення форми адміністратор має можливість переглянути попередній вигляд номеру або послуги, відредагувати або одразу опублікувати його. Система перевіряє заповненість обов'язкових полів і не

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		44

дозволить зберегти оголошення без них.

Опублікований об'єкт зберігається у базі даних та стає доступним на головній сторінці сайту, у відповідних категоріях та результатах пошуку.

Важливим елементом наповнення є підтримка актуальної інформації про додаткові послуги відпочинкового комплексу. До таких послуг можуть належати SPA-процедури, ресторанне обслуговування, конференц-зали, спортивні майданчики, басейни та розважальні заходи.

Для кожної послуги необхідно вказувати її опис, вартість та фотографії, що дозволить користувачу отримувати повну інформацію та приймати обґрунтовані рішення щодо їх замовлення.

Окрему увагу слід приділити регулярному оновленню та верифікації даних. Застарілі або вже неактуальні оголошення повинні видалятися або позначатися як «неактивні».

Це забезпечує якість і правдивість інформації на платформі, що є однією з головних переваг «Luxury Resort». Адміністратори можуть скористатися внутрішнім модулем звітності для перегляду стану активності кожного оголошення та отримання аналітики щодо переглядів, заявок, бронювань.

Важливою складовою процесу наповнення веб-додатку є робота з відгуками клієнтів та інформаційними матеріалами комплексу.

Це забезпечує постійне оновлення інформаційного наповнення ресурсу, покращує комунікацію з відвідувачами та сприяє підвищенню рівня довіри до відпочинкового комплексу «Luxury Resort».

Дотримання вищенаведених інструкцій дозволить забезпечити якісне представлення послуг, формувати довіру серед клієнтів та підтримувати репутацію сучасного, технологічного і відповідального відпочинкового комплексу «Luxury Resort», на ринку готельно-ресторанної справи.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		45

3.3 Інструкція з популяризації та підтримки вебзастосунку

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

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

Для залучення первинної аудиторії доцільно реалізувати програму лояльності з бонусною системою, яка передбачає пільгові умови отримання послуг, а також доступ до розширених статистик та аналітики. Крім того, важливо регулярно публікувати звіти про нові функції та оновлення платформи, демонструючи постійний розвиток і адаптацію до потреб ринку.

Ще одним напрямом популяризації є локальні рекламні кампанії у великих містах України – зовнішня реклама, участь у виставках і конференціях з готельно-ресторанного бізнесу. Також варто створити канал на YouTube з оглядами номерів, послуг, інтерв'ю з працівниками, туристичними порадами тощо. Важливою перевагою веб-додатку «Luxury Resort» є акцент на можливість бронювання, замовленням додаткових послуг та прозору інформацію, що варто активно просувати як конкурентну перевагу перед іншими платформами.

Окрім заходів з просування, велике значення має налагодження якісної

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		46

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

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

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

Отже, ефективне функціонування та популярність веб-додатку «Luxury Resort» на пряму залежить не лише від якості реалізації програмного забезпечення, але й від систематичної роботи з просуванням, підтримки та вдосконалення продукту. Завдяки комплексному підходу, який включає маркетингові активності, технічний супровід і постійне вдосконалення функціоналу, вебзастосунок може зайняти провідну позицію на українському ринку онлайн-сервісів з готельно-ресторанної справи.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		47

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини даного дипломного проєкту є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами, прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами.

Розрахунок вартості розробки виконується в декілька етапів:

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

4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами. Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 4.1.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дат		

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

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Планування та аналіз	Кер. проєкту Рm	8
		Інженер (І1)	8
2	Розробка технічного завдання	Кер. проєкту (Рm)	7
		Інженер (І1)	4
3	Дизайн інтерфейсу	Інженер (І1)	16
		Інженер (І2)	24
4	Розробка функціоналу	Інженер (І1)	48
5	Тестування та відладка	Тестувальник	8
6	Документування	Інженер (І1)	2
7	Розгортання та підтримка	Інженер (І2)	16
8	Управління проєктом	Кер. проєкту (Рm)	24
Разом			165

Сумарний час виконання операцій технологічного процесу становить 165 години.

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

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами.

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

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		49

підприємства.

Основна заробітна плата розраховується за формулою:

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

де: T_c – тарифна ставка, грн. (приймаємо для керівника проекту (Pm) – 450 грн./год, інженера (I2) – 272 грн./год.), інженера (I1) – 113 грн./год., тестувальник – 100 грн./год.; K_z – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

Керівника проекту (Pm) $Z_{\text{осн2}} = 39 * 450 = 17\,550$ грн.

Інженера (I2) $Z_{\text{осн4}} = 40 * 272 = 10\,880$ грн.

Інженера (I1) $Z_{\text{осн3}} = 78 * 113 = 8\,814$ грн.

Тестувальник $Z_{\text{осн4}} = 8 * 100 = 800$ грн.

Сумарна основна заробітна плата становить

$$Z_{\text{осн}} = 17\,550 + 10\,880 + 8\,814 + 800 = 38\,044 \text{ грн.}$$

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

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

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

Отже додаткова заробітна плата по категоріях працівників становить:

Керівника проекту $Z_{\text{дод2}} = 17\,550 * 0,1 = 1\,755$ грн.

Інженера (I2) $Z_{\text{дод3}} = 10\,880 * 0,1 = 1\,088$ грн.

Інженера (I1) $Z_{\text{дод4}} = 8\,814 * 0,1 = 881$ грн.

Тестувальник $Z_{\text{дод4}} = 800 * 0,1 = 80$ грн.

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод}} = 1\,755 + 1\,088 + 881 + 80 = 3\,804 \text{ грн.}$$

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

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

$$B_{\text{о.п.}} = 38\,044 + 3\,804 = 41\,848 \text{ грн.}$$

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

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

$$B_{\text{есв}} = 41\,848 * 0,22 = 9\,027 \text{ грн.}$$

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		50

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. 6 = 3+4+5
		Тарифна ставка, грн.	К-сть годин	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Кер. проекту (Pm)	450	39	17 550	1755	4 247	23 552
2	Інженера (I2)	272	40	10 880	1088	2 633	14 601
3	Інженера (I1)	113	78	8 814	881	2 133	11 827
4	Тестувальник	100	8	800	80	194	1 073
Разом				38 044	3 804	9 207	51 055

Отже, загальні витрати на оплату праці становлять 51 055 грн.

4.3. Розрахунок витрат на електроенергію

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

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

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

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

$$Z_{ек} = 0,82 * 165 * 15,94 = 2 157 \text{ грн.}$$

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		51

Витрати на електроенергію становлять 2 157 грн.

4.4. Розрахунок суми амортизаційних відрахувань розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами

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

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

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

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

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

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

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

Н_А – норма амортизації, 0,04 %.

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

$$A = \frac{50\,000,00 * 0,04}{150} * 165 = 2\,200 \text{ грн.}$$

4.5. Обчислення накладних витрат

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

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		52

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

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

де: H_B – накладні витрати.

$$H_B = 41\,848 * 0,4 = 16\,739 \text{ грн.}$$

4.6. Складання кошторису витрат та визначення собівартості розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами

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

Таблиця 4.4 - Кошторис витрат розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами

№	Зміст витрат	Сума, грн.	В % до загальної суми
1. 1	Витрати на оплату праці	51 055	71
2.	Витрати на електроенергію	2 157	3
3.	Амортизаційні відрахування	2 200	3
4.	Накладні витрати	16 739	23
5.	Собівартість	72 151	100

Собівартість (C_B) НДР розраховуємо за формулою:

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

Отже, собівартість дорівнює $C_B = 72\,151$ грн.

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		53

4.7. Розрахунок ціни розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка.

Ціну робіт можна визначити за формулою:

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

де: $C_{\text{в}}$ – собівартість; $P_{\text{рен}}$ – рівень рентабельності; ПДВ – ставка податку на додану вартість.

$$Ц = 72\,151 * (1 + 0,3) * (1 + 0,2) = 112\,556 \text{ грн.}$$

4.8. Визначення економічної ефективності і терміну окупності капітальних вкладень

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

Розрахунок прибутку визначається з а формулою:

$$П = Ц - C_{\text{в}} \quad (4.10)$$

де: $Ц$ – ціна розробки, грн.; $C_{\text{в}}$ – собівартість розробки, грн.

$$П = 112\,556 - 72\,151 = 40\,405 \text{ грн.}$$

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

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

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

$$\text{ЧТВ} = -72\,151 + \frac{40\,405}{(1+0,1)^1} + \frac{40\,405}{(1+0,1)^2} + \frac{40\,405}{(1+0,1)^3} = 28\,330 \text{ грн}$$

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		54

Якщо ЧТВ ≥ 0 , то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

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

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

$$T_{ок} = 2 + \frac{2027}{40\,405} = 2,1 \text{ р.}$$

Всі дані внесемо в зведену таблицю 4.5.

Таблиця 4.5 – Техніко-економічні показники розробки веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами

№ п/п	Показник	Значення
1.	Собівартість, грн.	72 151
2.	Плановий прибуток або грошовий потік, грн.	40 405
3.	Ціна, грн.	112 556
4.	Чиста теперішня вартість, грн.	28 330
5.	Термін окупності, рік	2,1

Прибутковість проекту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом 2.1 року. Отже, на основі отриманих показників можна зробити висновок, що розробка веб-додатку для відпочинкового комплексу «Luxury Resort» з готелем та додатковими послугами є доцільною з економічної точки зору. Узагальнені техніко-економічні показники розробки винесено в окремий аркуш графічної частини кваліфікаційної роботи.

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Кваліфікація виробничих травм

Кваліфікація виробничих травм є важливою складовою системи охорони праці, оскільки вона дозволяє визначити причини нещасних випадків, забезпечити соціальний захист працівників та запобігти подібним ситуаціям у майбутньому. Виробничий травматизм негативно впливає не лише на здоров'я працівників, але й на ефективність роботи підприємства, продуктивність праці та загальний стан трудової дисципліни. Саме тому питання безпеки праці регулюються законодавством України та перебувають під постійним контролем держави. Основними нормативними документами у цій сфері є Конституція України, Кодекс законів про працю України та Закон України «Про охорону праці». Конституція гарантує кожному працівникові право на безпечні та здорові умови праці, а законодавство покладає відповідальність за їх забезпечення на роботодавця.

Крім того, роботодавець зобов'язаний створювати належні умови праці, проводити навчання працівників і контролювати дотримання вимог безпеки на підприємстві. Відповідно до законодавства, нещасним випадком на виробництві вважається подія, що сталася під час виконання працівником трудових обов'язків і призвела до ушкодження здоров'я, втрати працездатності або смерті. До виробничих травм належать механічні ушкодження, опіки, ураження електричним струмом, отруєння, теплові удари та інші травми, отримані в процесі роботи. Такі випадки можуть виникати як через технічні несправності обладнання, так і через людський фактор або недотримання правил охорони праці. Основним завданням кваліфікації є визначення, чи пов'язаний нещасний випадок із виробництвом, а також встановлення причин його виникнення.

Це необхідно для правильного оформлення документації, проведення розслідування та забезпечення працівникові відповідних компенсацій і соціальних гарантій. У разі виробничої травми роботодавець зобов'язаний організувати розслідування події. Для цього створюється спеціальна комісія, яка проводить

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		56

огляд місця нещасного випадку, опитує свідків, аналізує технічний стан обладнання та перевіряє дотримання вимог охорони праці. На основі зібраних матеріалів комісія встановлює обставини події та складає відповідний акт. Важливим етапом є визначення того, чи виконував працівник свої службові обов'язки на момент отримання травми.

Також враховується, чи проходив працівник необхідні інструктажі, медичні огляди та чи був забезпечений засобами індивідуального захисту. Виробничі травми поділяються на легкі, тяжкі, групові та смертельні. Легкі травми не спричиняють тривалої втрати працездатності, тоді як тяжкі можуть призвести до інвалідності або тривалого лікування. Груповими вважаються випадки, у яких постраждало кілька працівників одночасно. Особливо небезпечними є смертельні випадки, які потребують спеціального розслідування за участю державних органів. Усі ці види травм мають важливе значення для статистичного обліку та аналізу рівня безпеки праці на підприємстві.

Основними причинами виробничого травматизму є порушення правил безпеки, несправність обладнання, недостатній контроль з боку керівництва, недотримання технологічних процесів та нехтування засобами індивідуального захисту. Досить часто причиною нещасних випадків стає неуважність працівників або недооцінка небезпеки під час виконання робіт. Для профілактики травматизму роботодавець повинен проводити інструктажі, навчання працівників, перевіряти технічний стан обладнання та створювати безпечні умови праці. Важливу роль також відіграє формування культури безпеки праці серед працівників та відповідального ставлення до власного здоров'я. У разі отримання виробничої травми працівник має право на соціальний захист і матеріальне забезпечення. Законодавство передбачає виплату допомоги у зв'язку з тимчасовою втратою працездатності, компенсацію витрат на лікування та страхові виплати у випадку інвалідності чи втрати працездатності. Таким чином, правильна кваліфікація виробничих травм має важливе значення для забезпечення безпеки працівників, встановлення причин нещасних випадків та вдосконалення системи охорони праці на підприємствах.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		57

5.2 Дії при виявленні пожежі

Дії при виявленні пожежі є важливою складовою системи охорони праці та пожежної безпеки, оскільки від правильності та швидкості реагування залежить життя і здоров'я людей, а також збереження майна та виробничих приміщень. Пожежі належать до найбільш небезпечних надзвичайних ситуацій, які можуть спричинити значні матеріальні збитки, руйнування будівель, забруднення навколишнього середовища та людські жертви.

Саме тому на кожному підприємстві повинні бути розроблені правила пожежної безпеки, плани евакуації та порядок дій працівників у разі виникнення пожежі. Законодавство України зобов'язує роботодавців забезпечувати належний протипожежний стан об'єктів, проводити інструктажі та навчання працівників, а також забезпечувати приміщення необхідними засобами пожежогасіння.

При виявленні пожежі або навіть незначних ознак загоряння, таких як запах диму, іскріння, підвищення температури чи поява відкритого вогню, працівник повинен діяти швидко та без паніки. Насамперед необхідно негайно повідомити про пожежу за номером 101, чітко назвавши адресу об'єкта, місце виникнення пожежі, наявність людей у приміщенні та можливу загрозу поширення вогню. Одночасно потрібно повідомити керівництво підприємства або відповідальну особу за пожежну безпеку. Швидке повідомлення пожежно-рятувальної служби дозволяє оперативно розпочати ліквідацію пожежі та зменшити ризик тяжких наслідків. Після повідомлення про пожежу необхідно організувати евакуацію людей із небезпечної зони відповідно до плану евакуації.

Особливу увагу слід приділяти людям, які можуть потребувати допомоги, зокрема дітям, людям похилого віку або особам з обмеженими можливостями. Під час евакуації забороняється користуватися ліфтами, оскільки у разі відключення електроенергії вони можуть зупинитися. Люди повинні рухатися до евакуаційних виходів швидко, але без паніки та тисняви. Якщо приміщення заповнене димом, рекомендується пересуватися пригнувшись або повзучи, оскільки біля підлоги концентрація диму є меншою. Якщо пожежа є невеликою та не становить

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		58

безпосередньої загрози життю, допускається використання первинних засобів пожежогасіння. До них належать вогнегасники, пожежні крани, пісок, вода або спеціальні покривала. При цьому необхідно враховувати характер загоряння, адже не всі речовини можна гасити водою. Наприклад, електрообладнання під напругою або легкозаймисті рідини не можна заливати водою через небезпеку ураження струмом або поширення полум'я.

Перед початком гасіння за можливості слід відключити електроживлення та зупинити роботу обладнання. Важливе значення має дотримання правил поведінки під час пожежі. Забороняється відкривати вікна та двері без потреби, оскільки приплив кисню може посилити горіння. Не можна повертатися до небезпечної зони за особистими речами чи документами, якщо це загрожує життю. У разі сильного задимлення рекомендується використовувати вологу тканину для захисту органів дихання. Якщо людина не може самостійно залишити приміщення, вона повинна привернути увагу рятувальників через вікно або інші доступні засоби зв'язку.

Після прибуття пожежно-рятувальних підрозділів необхідно повідомити їм всю відому інформацію про місце займання, можливі джерела небезпеки, наявність вибухонебезпечних чи токсичних речовин, а також про людей, які можуть залишатися у приміщенні. Працівники повинні виконувати вказівки рятувальників та не перешкоджати їхній роботі.

Після ліквідації пожежі проводиться розслідування причин її виникнення та оцінка завданих збитків.

Таким чином, правильні та своєчасні дії при виявленні пожежі мають надзвичайно важливе значення для збереження життя та здоров'я людей. Дотримання правил пожежної безпеки, регулярне проведення інструктажів і навчання працівників дозволяють знизити ризик виникнення пожеж та мінімізувати їх наслідки. Ефективна організація пожежної безпеки є необхідною умовою безпечної роботи будь-якого підприємства чи установи.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		59

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено повнофункціональний веб-додаток «Luxury Resort», призначений для автоматизації діяльності відпочинкового комплексу. У межах роботи було проведено аналіз існуючих рішень на ринку, визначено їх переваги та недоліки, що дозволило сформулювати технічні та функціональні вимоги до майбутнього програмного продукту.

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

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

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

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

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		60

ПЕРЕЛІК ПОСИЛАНЬ

- 8) Марціяш Г.Я., Слободян Р.О. Методичні вказівки до виконання дипломного проєкту для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки». Тернопіль : ВСП «ТФК ТНТУ ім. І. Пулюя», 2024, 48с.
- 9) OLX Недрухомість : вебсайт. URL: <https://www.olx.ua/nedvizhimost/> (дата звернення: 14.03.2025).
- 10) DOM.RIA – продаж і оренда нерухомості : вебсайт. URL: <https://dom.ria.com/> (дата звернення: 12.03.2025).
- 11) LUN.ua – агрегатор житла : вебсайт. URL: <https://www.lun.ua/> (дата звернення: 10.03.2025).
- 12) React – офіційна документація : вебсайт. URL: <https://react.dev/> (дата звернення: 11.03.2025).
- 13) TypeScript – офіційний сайт : вебсайт. URL: <https://www.typescriptlang.org/> (дата звернення: 13.03.2025).
- 14) Spring Boot Documentation : вебсайт. URL: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> (дата звернення: 15.03.2025).
- 15) PostgreSQL Documentation : вебсайт. URL: <https://www.postgresql.org/docs/> (дата звернення: 08.03.2025).
- 16) Docker – офіційна документація : вебсайт. URL: <https://docs.docker.com/> (дата звернення: 09.03.2025).
- 17) Material UI – компоненти для React : вебсайт. URL: <https://mui.com/> (дата звернення: 10.03.2025).
- 18) REST API Design – рекомендації Microsoft : вебсайт. URL: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design> (дата звернення: 13.03.2025).
- 19) GitHub – платформа для розробки : вебсайт. URL: <https://github.com/> (дата звернення: 07.03.2025).
- 20) Фленаган Д. *JavaScript*. Повне керівництво. 7-е вид. Київ: Видавництво «Діалектика», 2020. — 1072 с.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		61

21) Фаулер М. Шаблони корпоративних застосунків. Київ: Діалектика, 2021. – 650 с.

22) Еванс Е. Domain-Driven Design: Tackling Complexity in the Heart of Software. Boston: Addison-Wesley, 2004. — 560 р.

23) Вест Т. Docker для розробників: створення, тестування та розгортання застосунків із використанням контейнерів. Львів: Видавництво Старого Лева, 2023. — 312 с.

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		62

ДОДАТКИ

Додаток А. Лістинг файлу «pom.xml»

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.4.2</version>
    <relativePath/>
  </parent>

  <groupId>com.luxuryresort</groupId>
  <artifactId>luxury-resort-api</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>luxury-resort-api</name>
  <description>LuxuryResort diploma backend</description>

  <properties>
    <java.version>21</java.version>
    <!-- Newer Lombok: fewer javac glitches on non-21 JDKs; runtime still requires JDK
21 (see enforcer). -->
    <lombok.version>1.18.38</lombok.version>
    <mapstruct.version>1.6.3</mapstruct.version>
```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		63

```

<bucket4j.version>8.10.1</bucket4j.version>
<itext.version>8.0.5</itext.version>
<jjwt.version>0.12.6</jjwt.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
  </dependency>
  <dependency>
    <groupId>org.flywaydb</groupId>
    <artifactId>flyway-core</artifactId>

```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		64

```

</dependency>
<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-database-postgresql</artifactId>
</dependency>
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>org.mapstruct</groupId>
  <artifactId>mapstruct</artifactId>
  <version>${mapstruct.version}</version>
</dependency>
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
  <version>2.8.4</version>
</dependency>
<dependency>
  <groupId>com.itextpdf</groupId>
  <artifactId>kernel</artifactId>
  <version>${itext.version}</version>

```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		65

```

</dependency>
<dependency>
  <groupId>com.itextpdf</groupId>
  <artifactId>layout</artifactId>
  <version>${itext.version}</version>
</dependency>
<dependency>
  <groupId>com.bucket4j</groupId>
  <artifactId>bucket4j-core</artifactId>
  <version>${bucket4j.version}</version>
</dependency>
<dependency>
  <groupId>io.jsonwebtoken</groupId>
  <artifactId>jjwt-api</artifactId>
  <version>${jjwt.version}</version>
</dependency>
<dependency>
  <groupId>io.jsonwebtoken</groupId>
  <artifactId>jjwt-impl</artifactId>
  <version>${jjwt.version}</version>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>io.jsonwebtoken</groupId>
  <artifactId>jjwt-jackson</artifactId>
  <version>${jjwt.version}</version>
  <scope>runtime</scope>
</dependency>

```

```

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-test</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.testcontainers</groupId>
  <artifactId>junit-jupiter</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.testcontainers</groupId>
  <artifactId>postgresql</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-testcontainers</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>

<build>
  <plugins>

```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		67

```

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-enforcer-plugin</artifactId>
  <version>3.5.0</version>
  <executions>
    <execution>
      <id>enforce-jdk-21</id>
      <goals>
        <goal>enforce</goal>
      </goals>
      <configuration>
        <rules>
          <requireJavaVersion>
            <version>[21,22)</version>
            <message>
              Luxury Resort API must run and build with JDK 21 (Temurin 21).
              JDK 22+ compiles sometimes but Spring Boot 3.4.2 then fails at runtime
              (e.g. NoClassDefFoundError: List). Install JDK 21, set JAVA_HOME to it,
              verify: java -version
            </message>
          </requireJavaVersion>
        </rules>
        <fail>true</fail>
      </configuration>
    </execution>
  </executions>
</plugin>
<plugin>
  <groupId>org.springframework.boot</groupId>

```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		68

```

<artifactId>spring-boot-maven-plugin</artifactId>
<configuration>
  <excludes>
    <exclude>
      <groupId>org.projectlombok</groupId>
      <artifactId>lombok</artifactId>
    </exclude>
  </excludes>
</configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <configuration>
    <release>${java.version}</release>
    <annotationProcessorPaths>
      <path>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>${lombok.version}</version>
      </path>
      <path>
        <groupId>org.mapstruct</groupId>
        <artifactId>mapstruct-processor</artifactId>
        <version>${mapstruct.version}</version>
      </path>
      <path>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok-mapstruct-binding</artifactId>

```

```
<version>0.2.0</version>
</path>
</annotationProcessorPaths>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		70

Додаток Б. Лістинг файлу «.env.example»

SPRING_PROFILES_ACTIVE=dev

DATABASE_URL=jdbc:postgresql://localhost:5432/luxuryresort

DATABASE_USERNAME=postgres

DATABASE_PASSWORD=postgres

Якщо PostgreSQL піднятий через кореневий docker-compose.yml:

DATABASE_URL=jdbc:postgresql://localhost:5432/luxuryresort

DATABASE_USERNAME=resort_user

DATABASE_PASSWORD=resort_pass

Хмарна БД (Neon / Supabase / Railway тощо), без Docker локально:

DATABASE_URL=jdbc:postgresql://YOUR_HOST.aws.neon.tech/neondb?

sslmode=require

DATABASE_USERNAME=...

DATABASE_PASSWORD=...

SERVER_PORT=8080

CORS_ORIGINS=http://localhost:5173

JWT_SECRET=replace_with_at_least_32_characters_secret

					2026.КВР.122.421.03.00.00 ПЗ	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дат		

Додаток В. Лістинг файлу «Room.java»

```
package com.luxuryresort.domain.entity;

import com.luxuryresort.domain.enums.RoomStatus;
import com.luxuryresort.domain.enums.RoomType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.EnumType;
import jakarta.persistence.Enumerated;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import org.hibernate.annotations.JdbcTypeCode;
import org.hibernate.annotations.JdbcType;
import org.hibernate.dialect.PostgreSQLEnumJdbcType;
import org.hibernate.type.SqlTypes;

import java.math.BigDecimal;
import java.time.Instant;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		72

```

@Entity
@Table(name = "rooms")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class Room {

    @Id
    @GeneratedValue(strategy = GenerationType.UUID)
    private UUID id;

    @Column(nullable = false, length = 200)
    private String name;

    @Column(columnDefinition = "text")
    private String description;

    @Column(name = "base_price_per_night", nullable = false, precision = 10, scale = 2)
    private BigDecimal basePricePerNight;

    @Enumerated(EnumType.STRING)
    @JdbcType(PostgreSQLEnumJdbcType.class)
    @Column(name = "room_type", nullable = false, columnDefinition = "room_type")
    private RoomType roomType;

    @Column(name = "max_occupancy", nullable = false)
    private int maxOccupancy;

```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		73

```
@Column(name = "size_sqm", precision = 6, scale = 2)
```

```
private BigDecimal sizeSqm;
```

```
private Integer floor;
```

```
@Column(name = "room_number", nullable = false, unique = true, length = 10)
```

```
private String roomNumber;
```

```
@Enumerated(EnumType.STRING)
```

```
@JdbcType(PostgreSQLEnumJdbcType.class)
```

```
@Column(nullable = false, columnDefinition = "room_status")
```

```
private RoomStatus status;
```

```
@JdbcTypeCode(SqlTypes.JSON)
```

```
@Column(nullable = false, columnDefinition = "jsonb")
```

```
@Builder.Default
```

```
private List<String> amenities = new ArrayList<>();
```

```
@JdbcTypeCode(SqlTypes.JSON)
```

```
@Column(nullable = false, columnDefinition = "jsonb")
```

```
@Builder.Default
```

```
private List<String> images = new ArrayList<>();
```

```
@Column(name = "avg_rating", nullable = false, precision = 3, scale = 2)
```

```
private BigDecimal avgRating;
```

```
@Column(name = "review_count", nullable = false)
```

```
private int reviewCount;
```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		74

@Column(name = "created_at", nullable = false)

private Instant createdAt;

@Column(name = "updated_at", nullable = false)

private Instant updatedAt;

@Column(name = "deleted_at")

private Instant deletedAt;

}

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		75

Додаток Г. Лістинг файлу «Room.java»

```
package com.luxuryresort.application.service;

import com.luxuryresort.application.dto.request.RoomWriteRequest;
import com.luxuryresort.application.dto.response.PageResponse;
import com.luxuryresort.application.dto.response.RoomAvailabilityResponse;
import com.luxuryresort.application.dto.response.RoomResponse;
import com.luxuryresort.domain.enums.RoomType;
import org.springframework.data.domain.Pageable;

import java.math.BigDecimal;
import java.time.LocalDate;
import java.util.UUID;

public interface RoomService {

    PageResponse<RoomResponse> search(
        String q,
        RoomType type,
        BigDecimal priceMin,
        BigDecimal priceMax,
        Integer maxOccupancy,
        Boolean available,
        Pageable pageable
    );

    RoomResponse getById(UUID id);

    RoomAvailabilityResponse availability(UUID roomId, LocalDate checkIn, LocalDate
```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		76

checkOut);

RoomResponse create(RoomWriteRequest request);

RoomResponse update(UUID id, RoomWriteRequest request);

void softDelete(UUID id);

}

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		77

Додаток Д. Лістинг файлу «RoomController.java»

```
package com.luxuryresort.web.controller;

import com.luxuryresort.application.dto.request.RoomWriteRequest;
import com.luxuryresort.application.dto.response.PageResponse;
import com.luxuryresort.application.dto.response.RoomAvailabilityResponse;
import com.luxuryresort.application.dto.response.RoomResponse;
import com.luxuryresort.application.service.RoomService;
import com.luxuryresort.domain.enums.RoomType;
import com.luxuryresort.web.dto.ApiResponse;
import jakarta.validation.Valid;
import org.springframework.data.domain.Pageable;
import org.springframework.data.web.PageableDefault;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import java.math.BigDecimal;
import java.time.LocalDate;
import java.util.UUID;
```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		78

```

@RestController
@RequestMapping("/api/rooms")
public class RoomController {

    private final RoomService roomService;

    public RoomController(RoomService roomService) {
        this.roomService = roomService;
    }

    @GetMapping
    public ResponseEntity<ApiResponse<PageResponse<RoomResponse>>> list(
        @RequestParam(required = false) String q,
        @RequestParam(required = false) RoomType type,
        @RequestParam(required = false) BigDecimal priceMin,
        @RequestParam(required = false) BigDecimal priceMax,
        @RequestParam(required = false) Integer maxOccupancy,
        @RequestParam(required = false) Boolean available,
        @PageableDefault(size = 20) Pageable pageable
    ) {
        return ResponseEntity.ok(ApiResponse.ok(roomService.search(q, type, priceMin,
        priceMax, maxOccupancy, available, pageable)));
    }

    @GetMapping("/{id}")
    public ResponseEntity<ApiResponse<RoomResponse>> getById(@PathVariable UUID
    id) {
        return ResponseEntity.ok(ApiResponse.ok(roomService.getById(id)));
    }
}

```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		79

```

@GetMapping("/{id}/availability")
public ResponseEntity<ApiResponse<RoomAvailabilityResponse>> availability(
    @PathVariable UUID id,
    @RequestParam LocalDate checkIn,
    @RequestParam LocalDate checkOut
){
    return ResponseEntity.ok(ApiResponse.ok(roomService.availability(id, checkIn,
checkOut)));
}

@PostMapping
public ResponseEntity<ApiResponse<RoomResponse>> create(@Valid @RequestBody
RoomWriteRequest request) {
    return
ResponseEntity.status(HttpStatus.CREATED).body(ApiResponse.ok(roomService.create(
request)));
}

@PutMapping("/{id}")
public ResponseEntity<ApiResponse<RoomResponse>> update(
    @PathVariable UUID id,
    @Valid @RequestBody RoomWriteRequest request
){
    return ResponseEntity.ok(ApiResponse.ok(roomService.update(id, request)));
}

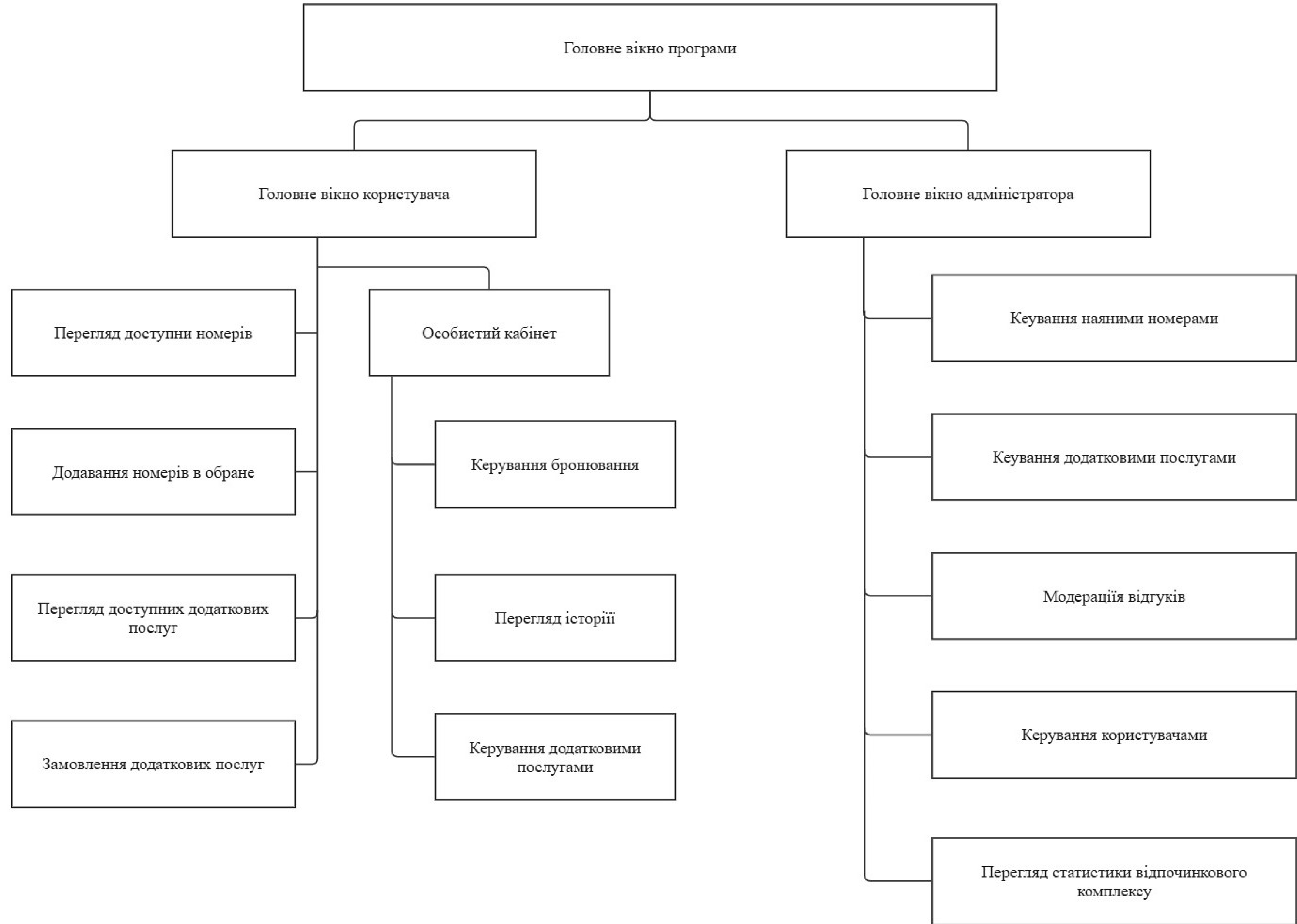
@DeleteMapping("/{id}")
public ResponseEntity<Void> delete(@PathVariable UUID id) {

```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		80

```
roomService.softDelete(id);  
return ResponseEntity.noContent().build();  
}  
}
```

					2026.KBP.122.421.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		81



Перш. застос. Стор. № Підп. і дата Зам. № № обліг. № обліг. Підп. і дата № № аркуш.

					2026.KBP.122.4213.00.00 CC		
Зм.	Арк.	№ док.	Підп.	Дата	Розробка вебзастосунку для відпочинкового комплексу «Luxury Resort». Схеми структурна вебсайту		
Розроб.		Борзичка Т.М.			Лист	Маса	Масштаб
Перев.		Марціш Г.Я.			Аркуш	Аркушів	1
І.контр.					ВСП ТФК ТНТЧ КН-421 м. Тернопіль		
Рецензент							
Н.контр.		Приймак В.А.					
Затв.							

Відпочинковий комплекс "Luxury Resort"

Гість
(Користувач)

- Вхід / Реєстрація
- Перегляд номерів
- Пошук та фільтрація номерів
- Перегляд додаткових послуг
- Бронювання номера
- Замовлення додаткових послуг
- Додавання номерів до обраного
- Перегляд історії бронювань
- Скасування бронювання
- Залишення відгуку

- Перегляд бронювань
- Підтвердження бронювань
- Зміна статусу бронювання
- Управління доступністю номерів
- Обробка заявок клієнтів
- Створення нового номера
- Редагування номерів
- Створення нової послуги
- Редагування послуг
- Створення нового користувача
- Управління користувачами
- Формування звітів
- Перегляд статистики бронювань
- Модерація відгуків

Працівник комплексу
(Ресепціоніст/
Менеджер)

Адміністратор

					2026.KBP.122.421.3.00.00 ДВ		
Зм.	Арк.	№ док.	Підп.	Дата	Розробка вебзастосунку для відпочинкового комплексу «Luxury Resort»		
Розроб.	Перев.	Морозюк Т.М.			Дизайн інтер'єру використання		
Т.контр.	Ресепціоніст	Пріймак В.А.			Аркцив	Аркцив	1
Н.контр.	Заліз.				ВСП ТФК ТНТЧ КН-421		
					м. Тернопіль		
					Формат А1		

Перш. застос.

Стор. №

Ліст. і дата

Ліст. і дата

Зем. №

Ліст. і дата

№

Перш. адм.с.

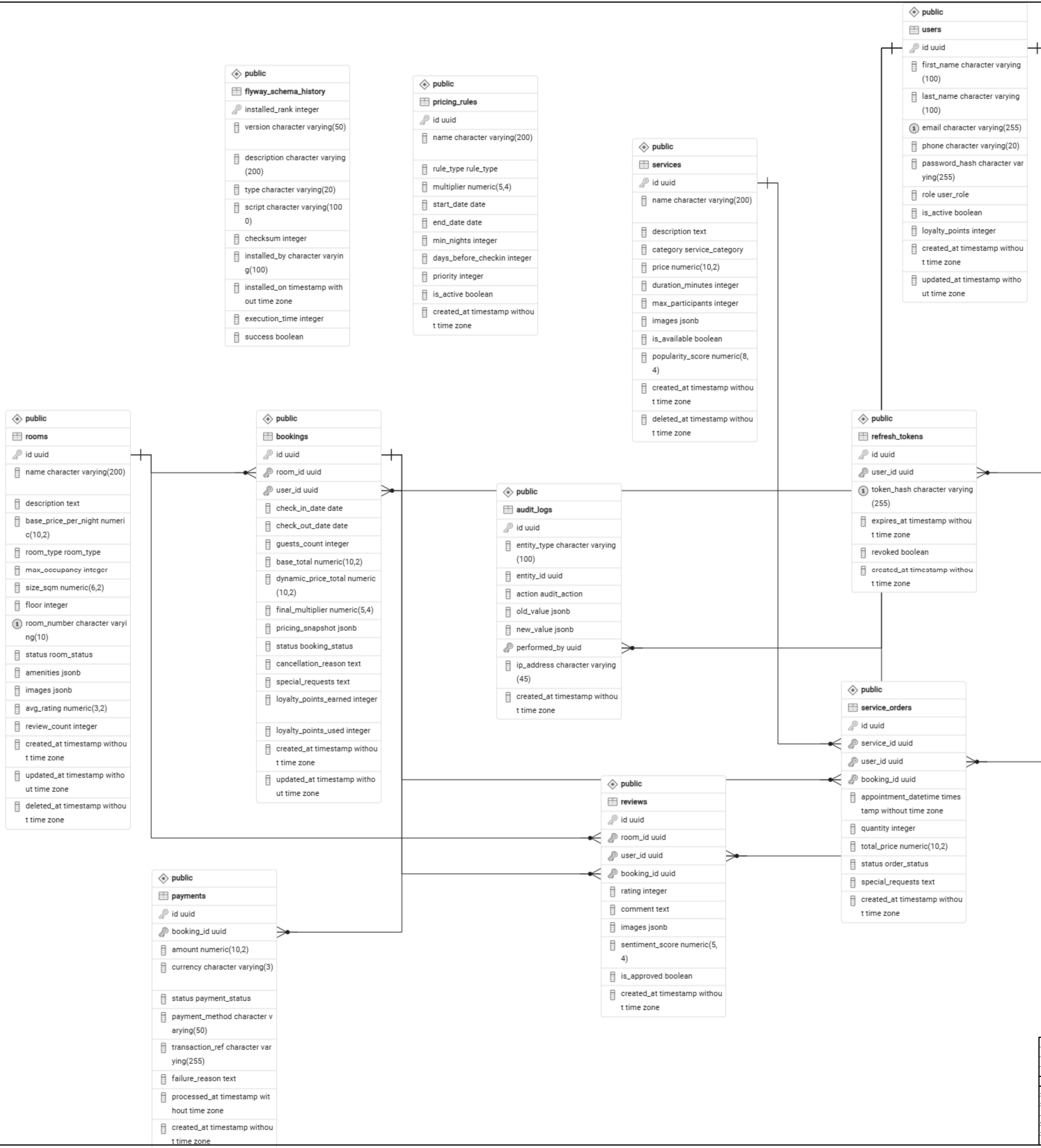
Ліст. і дата

Ліст. і дата

Ліст. і дата

Ліст. і дата

Ліст. і дата



2026.KBP.1224.213.00.00 БД			
Зм. Арк.	№ док.м.	Ліст.	Дата
Разроб	Бризицька Т.М.		
Левоб	Марціняк Г.Я.		
Текст			
Рецензент			
Нкантр.	Примак В.А.		
Затв.			
Розробка безпастосунку для відпочинкового комплексу «Сухигу Resort» ER-діаграма бази даних			
Літ.	Маса	Масштаб	
Архив	Архив	1	
ВЛП ТФХ ТНТУ КН-421			
м. Тернопіль			
Катюдаб			
Формат А1			

Таблиця техніко-економічних показників

№	Показник	Одиниці вимірювання	Значення
1	Мова програмування	–	Java, TypeScript
2	Технології клієнтської частини	–	HTML, CSS, React
3	Технології серверної частини	–	Spring Boot, Hibernate
4	Середовище розробки	–	Intelij IDEA, Web Storm
5	Загальний розмір програми	Мб	180
6	База даних	–	PostgreSQL
7	Міграція бази даних	–	Flyway
8	Собівартість проекту	грн	72 151
9	Плановий прибуток	грн	40 405
10	Ціна	грн	112 556
11	Чиста теперішня вартість	грн	28 330
12	Термін окупності	рік	2,1

2026.KBP.122.4213.00.00 ТБ				Лист	Масштаб
Вір. Лист	№ док.м.	Підп.	Дата	Розробка вебзастосунку	Лист
Розроб.	Боньська Т.М.			відпочинкового комплексу «Лихулу Resort»	Масштаб
Проб.	Марціниш Г.Я.			Таблиця техніко-економічних показників	Лист
Т.контр.					Листів 1
Н.контр.	Пріймак В.А.				ВСП ТФК ТНТУ КН-421
Утв.					м. Тернопіль