

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

бакалавра
(освітній рівень)

на тему: «Розробка функціоналу онлайн – бронювання та управління
готельним бізнесом»

Виконав: здобувач 4 року навчання

групи 4ПР1

напряму підготовки (спеціальності)

121 - «Інженерія програмного забезпечення»

(шифр і назва напряму підготовки, спеціальності)

Філінчук В.Р.

(підпис, прізвище та ініціали)

Керівник Доровська І.О.

(підпис, прізвище та ініціали)

Рецензент Григорова А.А.

(прізвище та ініціали)

Нормоконтролер Комісаров О.С.

(підпис, прізвище та ініціали)

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут, факультет, відділення *Інформаційних технологій та дизайну*
Кафедра, циклова комісія *Програмних засобів і технологій*
Освітньо-кваліфікаційний рівень *бакалавр*
Освітньо-професійна підготовка *Програмна інженерія*
(шифр і назва)
Спеціальність *121-Інженерія програмного забезпечення*
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

програмних засобів і технологій

О.Є. Огнєва

« » 2026 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Філіпчук Віктора Руслановича

(прізвище, ім'я, по батькові)

- Тема роботи «Розробка функціоналу онлайн – бронювання та управління готельним бізнесом»
керівник роботи доцент Доровська І.О.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджена наказом вищого навчального закладу від 16.01.2026 р. №82-С
- Строк подання студентом роботи 20.05.2026
- Вихідні дані до проекту (роботи) Розробка веб-додатку для готельного бізнесу з використанням php.
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
 - 1) огляд існуючих рішень;
 - 2) аналіз предметної області;
 - 3) проектування програмного продукту;
 - 4) розробка програмного продукту.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
 - 1) Діаграма Use Case основних сценаріїв використання системи;
 - 2) Концептуальна модель даних (ER-діаграма).
 - 3) Структура веб-сайту та навігація користувача.;
 - 4) Сторінка бронювання номерів;
 - 5) Результати тестування.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 15.10.2025**КАЛЕНДАРНИЙ
ПЛАН**

№	Назва етапів виконання роботи	Строк виконання етапів проекту (роботи)	Примітки
1	Отримання завдання	15.10.2025	Виконано
2	Підбір літератури	25.12.2025	Виконано
3	Аналіз предметної області	28.02.2026	Виконано
4	Розробка та обґрунтування завдання	07.02.2026	Виконано
5	Розробка концептуальної моделі	12.03.2026	Виконано
6	Розробка алгоритму	18.03.2026	Виконано
7	Проектування програми	25.03.2026	Виконано
8	Розробка інтерфейсу програми	30.04.2026	Виконано
9	Тестування програми	05.05.2026	Виконано
10	Оформлення пояснювальної записки	10.05.2026	Виконано
11	Захист кваліфікаційної роботи	22.06.2026	Виконано

Здобувач Філіпчук В.Р.

(підпис) (прізвище та ініціали)

Керівник проекту (роботи) Доровська І.О.

(підпис) (прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 58 сторінки, 9 рисунків, 28 джерел, 2 додатки.

Об'єкт дослідження - процеси онлайн-бронювання номерів та управління готельним бізнесом.

Предмет дослідження - методи та засоби розробки веб-додатку для автоматизації бронювання та управління готельними послугами.

Мета роботи - спроектувати та розробити веб-додаток для онлайн-бронювання номерів готелю, який забезпечує зручне бронювання без обов'язкової реєстрації, ефективне управління бронюваннями та аналітику.

У роботі проведено аналіз предметної області, визначено функціональні та нефункціональні вимоги, розроблено UML-діаграми, моделі бази даних та архітектуру системи. Для реалізації використано технології PHP, HTML5, CSS3, JavaScript та MySQL. Розроблено публічну частину сайту, адміністративну панель та проведено комплексне тестування.

Практична значущість роботи полягає у створенні готового веб-рішення, яке дозволяє невеликим готелям зменшити залежність від агрегаторів та отримати прямий канал взаємодії з клієнтами.

КЛЮЧОВІ СЛОВА: ВЕБ-ДОДАТОК, ОНЛАЙН-БРОНЮВАННЯ, ГОТЕЛЬНИЙ БІЗНЕС, PHP, MYSQL, БАЗА ДАНИХ, UML, АДМІНІСТРАТИВНА ПАНЕЛЬ.

АНОТАЦІЯ

Філіпчук В. Р. *Розробка функціоналу онлайн - бронювання та управління готельним бізнесом* - Кваліфікаційна робота бакалавра. Херсонський національний технічний університет, Хмельницький, 2026 р.

У роботі досліджено проблему автоматизації процесу бронювання номерів у готельному бізнесі. Проведено аналіз предметної області онлайн-бронювання, розглянуто існуючі рішення та визначено їх недоліки, зокрема високу комісію агрегаторів та обмежену гнучкість.

Спроектовано архітектуру веб-додатку: розроблено UML-діаграми варіантів використання, концептуальну та логічну моделі бази даних, обрано технологічний стек.

Реалізовано програмний продукт засобами PHP, HTML5, CSS3 та JavaScript з використанням локального сервера Open Server. Розроблено публічну частину сайту та адміністративну панель. Проведено функціональне та нефункціональне тестування веб-додатку.

В результаті виконання роботи створено повнофункціональний веб-додаток для онлайн-бронювання номерів готелю Bright Hotel, який забезпечує зручне бронювання без обов'язкової реєстрації, управління бронюваннями та аналітику. Практична значущість роботи полягає у створенні готового рішення, яке дозволяє невеликим готелям зменшити залежність від агрегаторів та отримати прямий канал взаємодії з клієнтами.

ABSTRACT

Filipchuk V. R. Development of online booking functionality and hotel business management — Bachelor's qualification thesis. Kherson National Technical University, Khmelnytskyi, 2026.

The thesis examines the problem of automating the hotel room booking process in the hospitality industry. The domain of online booking has been analyzed, existing solutions have been reviewed, and their disadvantages have been identified, particularly high aggregator commissions and limited flexibility.

The architecture of the web application was designed: UML use case diagrams, conceptual and logical database models were developed, and the technology stack was selected.

The software product was implemented using PHP, HTML5, CSS3, and JavaScript with the Open Server local environment. The public part of the website and the administrative panel were developed. Functional and non-functional testing of the web application was conducted.

As a result of the work, a fully functional web application for online hotel room booking "Bright Hotel" has been created. It provides convenient booking without mandatory registration, booking management, and analytics. The practical significance of the work lies in the development of a ready-made solution that allows small hotels to reduce dependence on aggregators and obtain a direct channel of interaction with customers.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Загальна характеристика предметної області онлайн-бронювання готельних послуг	9
1.2 Основні бізнес-процеси системи бронювання	10
1.3 Учасники процесу та їх роль у системі	11
1.4 Огляд існуючих рішень та проблеми, які вони не вирішують	12
Висновок до розділу 1	14
РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ ТА ПРОЄКТУВАННЯ СИСТЕМИ	16
2.1 Функціональні та нефункціональні вимоги до веб-додатку.....	16
2.2 Use Case діаграми основних сценаріїв використання.....	17
2.3 Структура веб-сайту та навігація користувача	19
Висновок до розділу 2	21
РОЗДІЛ 3 ПРОЄКТУВАННЯ БАЗИ ДАНИХ	23
3.1 Концептуальна модель даних (ER-діаграма).....	23
3.2 Логічна модель бази даних.....	24
3.2.1 Опис основних таблиць логічної моделі	26
3.3 Фізична реалізація бази даних (SQL-скрипти).....	27
Висновок до розділу 3	30
РОЗДІЛ 4 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ	32
4.1 Архітектура системи та особливості backend-частини.....	32
4.2 Розробка клієнтської частини	34
4.3 Реалізація основного функціоналу системи бронювання.....	45
4.4 Розробка адміністративної панелі.....	48

4.5 Тестування веб-додатку та аналіз результатів	51
4.6 Інструкція користувача.....	52
Висновок до розділу 4	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57
ДОДАТОК А.....	59
ДОДАТОК В	61

ВСТУП

Сучасний розвиток готельної галузі характеризується стрімким переходом від традиційних методів бронювання до цифрових онлайн-систем. Зростання кількості подорожей та підвищення вимог клієнтів до швидкості й зручності обслуговування зумовили необхідність створення ефективних веб-рішень для автоматизації процесу бронювання номерів. Багато невеликих і середніх готелів досі використовують застарілі методи управління бронюваннями (телефонні дзвінки, електронна пошта, таблиці Excel), що призводить до помилок, втрати потенційних клієнтів та зниження конкурентоспроможності.

У зв'язку з цим було прийнято рішення про розробка функціоналу онлайн-бронювання та управління готельним бізнесом. Оптимальним технологічним рішенням для реалізації поставленого завдання обрано зв'язку мови програмування PHP з технологіями HTML, CSS та JavaScript. Такий вибір обумовлений широкою поширеністю цих технологій, їх надійністю, відносною простотою підтримки та можливістю розгортання на більшості хостингів без значних фінансових витрат.

Для створення якісного програмного продукту в процесі роботи необхідно було вирішити низку важливих завдань: провести детальний аналіз предметної області онлайн-бронювання, спроектувати зручну та інтуїтивно зрозумілу структуру веб-сайту, розробити ефективну базу даних, реалізувати клієнтську та серверну частини додатку, створити функціональну адміністративну панель, а також забезпечити належний рівень тестування системи.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна характеристика предметної області онлайн-бронювання готельних послуг

Предметна область онлайн-бронювання готельних послуг охоплює сукупність процесів, пов'язаних зі збиранням, обробкою, зберіганням і використанням інформації про номерний фонд, доступність номерів, тарифах та взаємодії між готелем і клієнтами. Сучасні системи онлайн-бронювання є важливою складовою цифрової інфраструктури готельного бізнесу [19], оскільки дозволяють автоматизувати прийом і обробку заявок, підвищити прозорість послуг та забезпечити клієнтам зручний цілодобовий доступ до інформації [15].

Онлайн-бронювання дає змогу потенційним гостям самостійно здійснювати пошук, вибір та резервування номерів через веб-інтерфейс без прямого контакту з персоналом готелю. Ключовими елементами предметної області є підтримка актуальності даних про наявність номерів у реальному часі, розрахунок вартості проживання з урахуванням дат, типу номеру та додаткових послуг, а також забезпечення безпечної обробки персональних даних клієнтів [4].

Важливою особливістю сучасних систем є їх багатокomпонентність. Вони включають модулі авторизації користувачів, каталог номерів, календар доступності, систему тарифів і знижок, модуль бронювання, адміністративну панель та механізми взаємодії з клієнтами. Крім того, такі системи часто інтегруються з платіжними шлюзами, системами управління готелем (PMS) та зовнішніми сервісами [7].

Особливе значення в предметній області має баланс інтересів двох основних груп користувачів. Для клієнтів пріоритетом є зручний інтерфейс, швидкість роботи, прозорість цін і можливість керувати своїми бронюваннями. Для персоналу готелю критично важливими є інструменти контролю завантаженості номерного фонду, управління статусами бронювань, обробки змін і скасування, а також отримання аналітичних даних для прийняття управлінських рішень [10].

Сучасні системи онлайн-бронювання повинні відповідати високим вимогам щодо захисту персональних даних, стабільності роботи, адаптивності інтерфейсу

та доступності відповідно до стандартів WCAG [22]. Також важливим є забезпечення гнучкості системи - можливості швидкого внесення змін у тарифи, акції, правила бронювання та політику скасування [18].

Таким чином, предметна область онлайн-бронювання готельних послуг являє собою складну динамічну систему, яка поєднує технічні, організаційні, правові та маркетингові аспекти. Успішна реалізація веб-додатку в цій сфері вимагає комплексного підходу до аналізу бізнес-процесів, проєктування архітектури та забезпечення високого рівня безпеки й зручності використання [9, 21].

1.2 Основні бізнес-процеси системи бронювання

Основні бізнес-процеси системи онлайн-бронювання утворюють єдиний цикл, що забезпечує взаємодію клієнта з готелем на всіх етапах - від пошуку номера до заселення та післяпродажного обслуговування.

Центральним процесом є пошук і вибір номера. Клієнт задає бажані дати проживання, кількість гостей та інші параметри, після чого система в реальному часі перевіряє доступність номерів, враховуючи їх тип, категорію, технічний стан та діючі тарифи. Результатом є формування актуальної пропозиції з урахуванням можливих знижок і додаткових послуг.

Наступним етапом є оформлення бронювання. На цьому етапі відбувається введення персональних даних клієнта, розрахунок остаточної вартості, застосування промокодів та підтвердження умов бронювання. Система виконує автоматичну валідацію даних і блокує номер на вказаний період для запобігання подвійному бронюванню.

Важливим бізнес-процесом є обробка оплати. Система забезпечує інтеграцію з платіжними сервісами, фіксацію статусу платежу, формування підтверджувальних документів і передачу інформації до бухгалтерського обліку. Після оформлення бронювання запускається процес управління замовленнями, який включає можливість зміни або скасування бронювання (відповідно до правил

готелю), зміну статусів (підтверджено, гарантовано, заїзд, виїзд тощо) та ведення історії взаємодії з клієнтом.

Значну роль відіграє аналітичний процес. Система накопичує дані про завантаженість номерного фонду, популярність певних категорій номерів, конверсію бронювань, середній чек і джерела трафіку. Отримана аналітика використовується керівництвом готелю для коригування цін, планування акцій та оптимізації роботи.

Не менш важливим є процес комунікації з клієнтом - автоматична відправка підтверджень, нагадувань, інформаційних повідомлень та опитувань щодо якості обслуговування. Таким чином, бізнес-процеси системи онлайн-бронювання охоплюють повний цикл управління номерним фондом і взаємовідносинами з клієнтами. Їх ефективна автоматизація дозволяє суттєво підвищити якість обслуговування, зменшити операційні витрати та підвищити конкурентоспроможність готельного підприємства.

1.3 Учасники процесу та їх роль у системі

Учасниками процесу онлайн-бронювання готелю Bright Hotel є суб'єкти, які безпосередньо взаємодіють із системою або залучені до бізнес-процесів бронювання. Правильне визначення ролей і розмежування прав доступу є ключовим для забезпечення безпеки, зручності використання та ефективного управління системою [9].

Основним зовнішнім учасником є клієнт (потенційний гість). Він має доступ до публічної частини веб-додатку і може виконувати такі дії: переглядати інформацію про готель та номерний фонд, здійснювати пошук доступних номерів за заданими датами, ознайомлюватися з тарифами та додатковими послугами, оформлювати бронювання, реєструватися в особистому кабінеті, переглядати історію своїх замовлень, вносити зміни в бронювання та скасовувати його (у межах встановлених правил). Клієнт також може залишати відгуки та отримувати автоматичні повідомлення від системи [15].

Головним внутрішнім учасником є адміністратор готелю (менеджер). Він працює в адміністративній панелі системи і має значно ширші права. До його функцій належать: перегляд і управління всіма бронюваннями, зміна статусів замовлень, редагування інформації про номери (доступність, ціни, опис, фотографії), управління тарифами та акціями, обробка платежів, підтвердження або відхилення бронювань, робота з клієнтською базою даних, генерація звітів про завантаженість готелю та фінансові показники. Адміністратор також може створювати та редагувати контент сайту (новини, акції, фото галерею) [10].

Окрім людських учасників, важливим елементом є сам веб-додаток, який виступає активним учасником багатьох процесів. Система автоматично виконує перевірку доступності номерів у реальному часі, блокує номери при оформленні бронювання, розраховує вартість з урахуванням знижок, надсилає підтвердження та нагадування клієнтам, веде журнал подій і забезпечує синхронізацію даних між клієнтською та адміністративною частинами [7].

Взаємодія між учасниками відбувається за чітко визначеними сценаріями. Клієнт взаємодіє переважно з фронтендом системи, тоді як адміністратори - з захищеною адміністративною панеллю. При цьому всі критичні дії (зміна статусу бронювання, редагування цін, підтвердження оплати) фіксуються в журналі системи, що забезпечує прозорість і можливість подальшого аудиту [6].

Таким чином, чітке розмежування ролей учасників процесу дозволяє забезпечити необхідний рівень безпеки, зручності та ефективності роботи системи, а також є основою для подальшого проєктування архітектури веб-додатку та моделювання його функціональних можливостей [9, 21].

1.4 Огляд існуючих рішень та проблеми, які вони не вирішують

На сучасному ринку представлено значну кількість рішень для онлайн-бронювання готельних послуг. Усі їх можна умовно поділити на три основні групи: глобальні агрегатори, спеціалізовані хмарні системи управління готелями (PMS) та самостійні веб-сайти готелів .

До найбільш відомих глобальних платформ належать Booking.com, Hotels.com, Expedia та Airbnb. Ці сервіси володіють потужною аудиторією, розвиненою системою пошуку та високим рівнем довіри клієнтів. Однак їх використання має суттєві недоліки для власників готелів. По-перше, комісія за бронювання зазвичай становить від 10% до 25% від вартості номера. По-друге, готель фактично втрачає прямий контакт з клієнтом, оскільки всі комунікації та дані про гостя залишаються на платформі агрегатора. По-третє, алгоритми ранжування часто ставлять менші готелі в не вигідне становище порівняно з великими мережами [14, 19].

Спеціалізовані системи управління готелями (Property Management Systems), такі як Cloudbeds, HotelRunner, SiteMinder, Oracle Hospitality та Opera, пропонують комплексне рішення для управління номерним фондом, бронюваннями, каналами продажів та звітністю. Перевагою таких систем є потужна функціональність і інтеграція з великою кількістю каналів збуту [15]. Водночас більшість з них є платними, мають складну структуру, вимагають регулярної абонентської плати та спеціальних знань для роботи. Для невеликих і середніх готелів такі рішення часто виявляються економічно недоцільними через високу вартість впровадження та підтримки.

Багато готелів також використовують власні веб-сайти, розроблені на основі CMS (WordPress, Joomla, OpenCart). Такі рішення зазвичай дешевші, але мають обмежену функціональність: слабку систему реального часу перевірки доступності номерів, відсутність надійного механізму блокування номерів, примітивну адміністративну панель та низький рівень захисту даних.

Аналіз існуючих рішень дозволяє виділити низку проблем, які вони повністю або частково не вирішують, особливо для невеликих готелів:

- Висока вартість користування та комісійні платежі, що суттєво зменшують прибуток закладу.
- Відсутність повного контролю над клієнтською базою даних та історією бронювань.
- Недостатня гнучкість у налаштуванні тарифів, акцій та правил бронювання під специфіку конкретного готелю.

- Складність або неможливість швидкого внесення змін в інтерфейс та функціональність без залучення сторонніх розробників.
- Низький рівень персоналізації сервісу для постійних клієнтів.
- Проблеми з інтеграцією та синхронізацією даних між різними каналами бронювання в реальному часі.
- Недостатня увага до локальних особливостей (мова, валюта, специфіка законодавства, популярні способи оплати в Україні).

Особливо гостро ці проблеми проявляються у невеликих сімейних та бутік-готелях, до яких належить і Bright Hotel. Використання лише агрегаторів призводить до втрати значної частини маржі, а впровадження потужних PMS-систем - до необґрунтовано високих витрат [26].

Таким чином, розробка власної системи онлайн-бронювання дозволяє готелю Bright Hotel усунути зазначені недоліки: отримати прямий канал взаємодії з клієнтами, суттєво знизити комісійні витрати, повністю контролювати дані, забезпечити високу гнучкість налаштувань та адаптувати систему *precisely* під потреби конкретного закладу. Саме тому створення індивідуального веб-додатку є актуальним і економічно виправданим рішенням [20].

Висновок до розділу 1

У першому розділі дипломної роботи проведено комплексний аналіз предметної області онлайн-бронювання готельних послуг. Було встановлено, що сучасна система онлайн-бронювання є багатокомпонентною динамічною системою, яка поєднує технічні, організаційні, економічні та правові аспекти діяльності готельного підприємства.

Розглянуто основні бізнес-процеси, що охоплюють повний цикл взаємодії клієнта з готелем - від пошуку та вибору номера до оформлення бронювання, оплати, управління замовленнями та аналітики. Визначено ключових учасників процесу та їх ролі в системі, а саме: клієнта (гостя), адміністратора готелю та суперадміністратора, кожен з яких має чітко розмежовані права доступу та функціональні можливості.

Аналіз існуючих рішень показав, що попри значну кількість глобальних агрегаторів (Booking.com, Hotels.com) та спеціалізованих PMS-систем, вони мають суттєві недоліки для невеликих і середніх готелів. Найважливішими з них є високий рівень комісійних платежів, втрата прямого контакту з клієнтом, обмежена гнучкість налаштувань під специфіку конкретного закладу, а також значні фінансові витрати на використання потужних комерційних рішень.

Таким чином, актуальність розробки власної системи онлайн-бронювання для готелю Bright Hotel зумовлена необхідністю створення сучасного, економічно ефективного, гнучкого та повністю контрольованого інструменту, який дозволить оптимізувати управління номерним фондом, підвищити якість обслуговування клієнтів, зменшити операційні витрати та зміцнити конкурентні позиції готелю на ринку.

РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ ТА ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Функціональні та нефункціональні вимоги до веб-додатку

Важливим етапом розробки будь-якого програмного забезпечення є чітке формулювання функціональних та нефункціональних вимог [6]. У рамках дипломної роботи вимоги до веб-додатку для онлайн-бронювання номерів готелю Bright Hotel були визначені на основі проведеного аналізу предметної області та бізнес-процесів.

Функціональні вимоги визначають основні можливості системи для різних категорій користувачів. До них належить реалізація публічної частини веб-додатку, яка забезпечує ознайомлення відвідувачів з інформацією про готель, послуги та номерний фонд. Система повинна надавати можливість здійснення детального пошуку та фільтрації номерів за заданими датами, кількістю гостей, типом розміщення та ціновим діапазоном. Важливим є відображення детальної інформації про кожен номер, включаючи фотографії, опис, зручності та актуальну вартість. Ключовою функціональною можливістю є процес онлайн-бронювання з автоматичною перевіркою доступності номерів у реальному часі [18].

Окрім цього, система повинна підтримувати управління контентом і бронюваннями через адміністративну панель. Адміністратор має отримувати можливість редагування інформації про номери, тарифи, акції та галерею, зміни статусів бронювань, обробки платежів, а також генерації аналітичних звітів щодо завантаженості готелю та фінансових показників. Передбачається також реалізація автоматичної комунікації з клієнтами шляхом відправки електронних повідомлень (підтвердження бронювання, нагадування, інформація про скасування) [15].

Нефункціональні вимоги відображають якісні характеристики майбутнього веб-додатку. Інтерфейс системи повинен бути адаптивним і коректно відображатися на різних типах пристроїв [21]. Час відповіді на основні операції (пошук номерів, оформлення бронювання) не повинен перевищувати трьох секунд. Особлива увага приділяється питанням безпеки - система має забезпечувати надійний захист персональних даних клієнтів, розмежування прав доступу та протидію несанкціонованому доступу [22]. Важливими є також

стабільність роботи при багатокористувацькому доступі, інтуїтивність інтерфейсу, сучасний дизайн та відповідність базовим стандартам веб-доступності.

Крім того, система повинна бути зручною в подальшій підтримці та масштабуванні, а також забезпечувати можливість регулярного резервного копіювання даних і швидкого відновлення після можливих збоїв [9].

2.2 Use Case діаграми основних сценаріїв використання

Для моделювання функціональної поведінки веб-додатку було використано UML-діаграми варіантів використання (Use Case). У системі виділено два основних актори: Гість та Адміністратор.

Актор Гість має доступ до публічної частини сайту і може виконувати такі дії, як перегляд головної сторінки, перегляд номерів, створення бронювання, перегляд інформації про готель та блогу.

Актор Адміністратор працює в адміністративній панелі і здійснює керування бронюваннями, номерами, контентом сайту, аналітикою та повідомленнями див. рис. 2.1.



Рисунок 2.1 - Діаграма Use Case основних сценаріїв використання веб-додатку
Bright Hotel

Представлена діаграма наочно демонструє розмежування функціональних можливостей між публічною та адміністративною частинами системи. Актор «Гість» має доступ лише до інформаційних та бронювальних функцій, тоді як актор «Адміністратор» відповідає за управління всіма ключовими аспектами роботи веб-додатку.

Використання UML-діаграм Use Case дозволило забезпечити повноту функціонального покриття системи відповідно до виявлених у першому розділі бізнес-процесів. Побудована модель стала важливою основою для подальшого

проектування структури веб-сайту, бази даних та безпосередньої програмної реалізації [12].

2.3 Структура веб-сайту та навігація користувача

Структура веб-додатку Bright Hotel побудована за принципом чіткого розмежування публічної та адміністративної частин. Важливою особливістю системи є можливість бронювання номерів без обов'язкової реєстрації клієнта, що спрощує процес взаємодії з користувачами та підвищує доступність сервісу [21].

Публічна частина веб-сайту доступна всім відвідувачам без авторизації і складається з таких основних сторінок: головна сторінка, «Номери», «Про готель», «Контакти», «Блог» та сторінка бронювання. Навігація в публічній частині реалізована за допомогою горизонтального меню, що є стандартним рішенням для сучасних веб-додатків [1, 26].

На головній сторінці розміщено блок швидкого пошуку номерів, який дозволяє користувачу вказати дати заїзду та виїзду і перейти до перегляду доступних варіантів. Оформлення бронювання здійснюється через спеціальну форму, в яку клієнт вносить свої контактні дані (ПІБ, телефон, email), інформацію про дати проживання та додаткові побажання. Після відправлення форми заявка надходить до адміністративної панелі.

Адміністративна частина системи захищена механізмом авторизації і включає такі основні розділи: авторизація адміністратора, керування бронюваннями, керування номерами, перегляд аналітики, керування повідомленнями користувачів та керування контентом сайту. Навігація в адміністративній панелі реалізована через бічне меню, що забезпечує швидкий доступ до всіх модулів управління [7].

Відсутність особистого кабінету клієнта та можливості перегляду власних бронювань без реєстрації обумовлює спрощену архітектуру публічної частини, орієнтовану на швидке одноразове бронювання див. рис. 2.2.

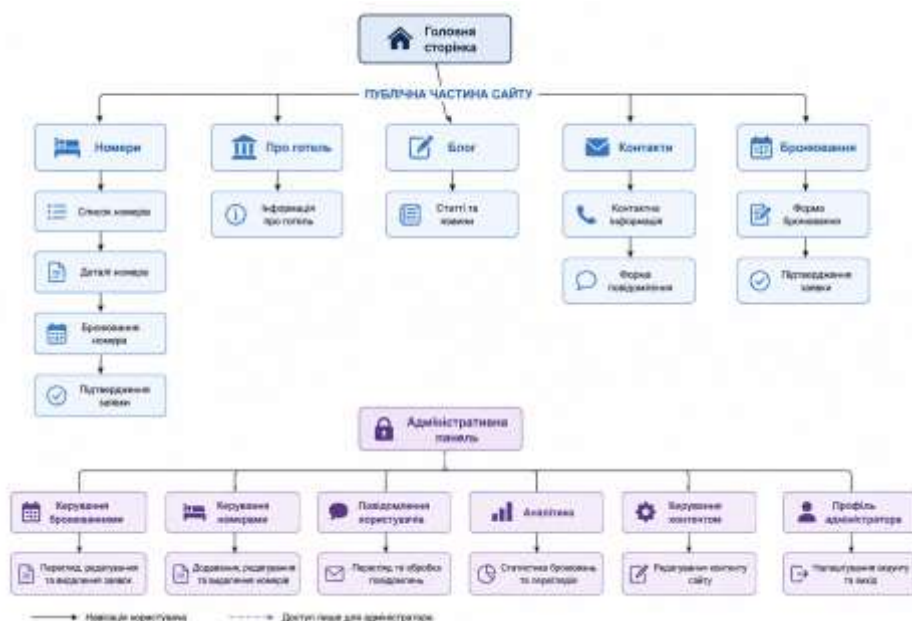


Рисунок 2.2 - Структура веб-сайту та навігація користувача Bright Hotel

Така структура веб-додатку забезпечує баланс між простотою використання для клієнтів та повноцінними інструментами управління для адміністратора. Відсутність реєстрації для гостей дозволяє зменшити кількість кроків при бронюванні, що позитивно впливає на конверсію, тоді як розвинена адміністративна панель дає можливість ефективно обробляти всі заявки та керувати контентом сайту [18].

2.4 Вибір технологічного стеку та інструментів розробки

Вибір технологічного стеку є одним із ключових етапів розробки веб-додатку, оскільки від нього залежить швидкість створення системи, її продуктивність, масштабованість та зручність подальшої підтримки.

Для реалізації веб-додатку Bright Hotel було обрано наступний технологічний стек:

Серверна частина (Backend) розроблена з використанням мови програмування PHP. Вибір PHP обумовлений кількома факторами. По-перше, це одна з найбільш поширених серверних мов для створення динамічних веб-сайтів, яка добре підходить для роботи з базами даних [21]. По-друге, PHP має велику кількість готових рішень, високу швидкість розробки та відмінну сумісність з більшістю хостинг-провайдерів [24]. Крім того, PHP дозволяє ефективно

реалізовувати основні бізнес-процеси бронювання, такі як обробка форм, перевірка доступності номерів та взаємодію з базою даних.

Клієнтська частина (Frontend) реалізована за допомогою стандартних веб-технологій: HTML5, CSS3 та JavaScript. HTML5 забезпечує сучасну семантичну розмітку сторінок, CSS3 використовується для створення адаптивного дизайну та візуального оформлення, а JavaScript відповідає за інтерактивність інтерфейсу (валідацію форм, динамічне оновлення даних, роботу з календарем бронювання тощо) [16]. Відмова від складних JavaScript-фреймворків (типу React або [Vue.js](https://vuejs.org/) [28]) пояснюється відносно невеликим обсягом проєкту та бажанням забезпечити максимальну простоту коду та швидкість завантаження сторінок.

Для локальної розробки та тестування системи використовувалося середовище Open Server, яке містить усе необхідне ПЗ (Apache, PHP, MySQL, phpMyAdmin) в одному пакеті. Open Server забезпечує зручне налаштування віртуальних хостів, швидке розгортання проєкту та зручну роботу з базою даних [19].

У якості основного редактора коду обрано Visual Studio Code. Цей редактор вирізняється високою швидкістю роботи, великою кількістю розширень, підтримкою PHP, JavaScript, інтеграцією з Git та зручними інструментами налагодження. Вибраний технологічний стек повністю відповідає поставленим функціональним і нефункціональним вимогам та дозволяє ефективно реалізувати всі необхідні модулі системи онлайн-бронювання [7, 9].

Висновок до розділу 2

У даному розділі здійснено постановку задачі та проєктування системи онлайн-бронювання номерів готелю Bright Hotel.

Було сформульовано функціональні та нефункціональні вимоги до веб-додатку, які визначають необхідний набір можливостей для клієнтів та адміністраторів. Особлива увага приділялася спрощеному процесу бронювання без обов'язкової реєстрації користувача, що відповідає специфіці малого та середнього готельного бізнесу.

За допомогою UML-діаграм Use Case проведено моделювання основних сценаріїв використання системи, чітко визначено двох акторів (Гість та Адміністратор) та їх функціональні можливості. Це дозволило формалізувати межі системи та забезпечити відповідність її поведінки виявленим бізнес-процесам.

Також було розроблено структуру веб-сайту та навігацію користувача. Архітектура системи передбачає чітке розмежування публічної частини, доступної всім відвідувачам, та захищеної адміністративної панелі. Відсутність особистого кабінету клієнта обумовлює спрощений та зручний шлях бронювання.

У розділі обґрунтовано вибір технологічного стеку, що включає мову програмування PHP, технології HTML, CSS, JavaScript, середовище розробки Visual Studio Code та локальний сервер Open Server. Даний стек забезпечує оптимальне співвідношення швидкості розробки, продуктивності, безпеки та зручності подальшої підтримки системи.

РОЗДІЛ 3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ

3.1 Концептуальна модель даних (ER-діаграма)

Концептуальне моделювання є ключовим етапом проектування бази даних, що дозволяє представити основні сутності предметної області та взаємозв'язки між ними в абстрактній формі, незалежно від конкретної системи управління базами даних [17].

Була розроблена концептуальна модель даних веб-додатку Bright Hotel у вигляді ER-діаграми (Entity-Relationship Diagram), виконаної в нотації Crow's Foot. Дана модель відображає основні сутності системи та встановлені між ними зв'язки див. рис. 3.1.

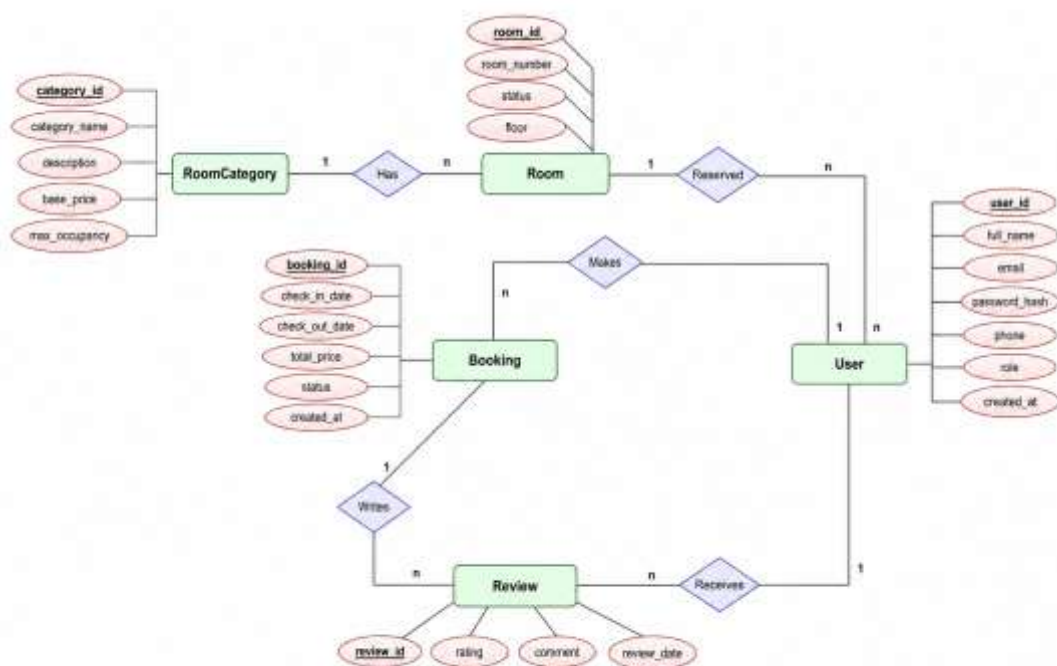


Рисунок 3.1 – Концептуальна модель даних (ER-діаграма) веб-додатку Bright Hotel

Центральне місце в моделі посідає сутність Booking (Бронювання), яка є ключовим елементом системи онлайн-бронювання. Кожне бронювання пов'язане зі сутністю Room (Номер), оскільки в системі фіксується, який саме номер заброньовано на конкретні дати. Один номер може мати кілька бронювань у різний час, що відображається зв'язком один-до-багатьох.

Сутність Room пов'язана зі сутністю RoomCategory (Категорія номерів). Одна категорія може включати кілька номерів (зв'язок один-до-багатьох). До атрибутів категорії належать назва, опис, базова ціна та максимальна кількість гостей, що дозволяє ефективно керувати типами номерів і їх ціноутворенням.

Сутність User (Користувач) представляє всіх користувачів системи, включаючи як адміністраторів, так і клієнтів, які здійснюють бронювання. Кожен користувач може створювати кілька бронювань, що реалізовано відповідним зв'язком один-до-багатьох. Крім того, користувач може залишати відгуки про готель.

Сутність Review (Відгук) пов'язана зі сутностями User та Booking. Такий подвійний зв'язок дозволяє забезпечити, щоб відгук залишався тільки після реального бронювання, а також прив'язувати відгук до конкретного клієнта. Це підвищує достовірність відгуків у системі.

Усі сутності містять необхідний набір атрибутів. Наприклад, сутність Booking включає дати заїзду та виїзду, загальну вартість проживання, статус бронювання та дату створення заявки. Сутність Room містить інформацію про номер кімнати, поверх і поточний статус. Така структура даних дозволяє підтримувати актуальну інформацію про завантаженість номерного фонду в реальному часі [12]. Побудована концептуальна модель повністю відображає специфіку предметної області онлайн-бронювання, забезпечує цілісність і непротиворіччя даних, а також створює надійну основу для розробки логічної та фізичної моделі бази даних [17].

3.2 Логічна модель бази даних

Логічна модель бази даних розроблена на основі концептуальної ER-діаграми та є подальшим кроком у конкретизації структури даних. На цьому етапі були визначені назви таблиць, склад атрибутів, типи даних для кожного поля, первинні та зовнішні ключі, а також обмеження цілісності. Перехід до логічної моделі дозволив врахувати особливості реалізації в реляційній системі управління

базами даних MySQL та підготувати основу для створення фізичних SQL-скриптів див. рис. 3.2.

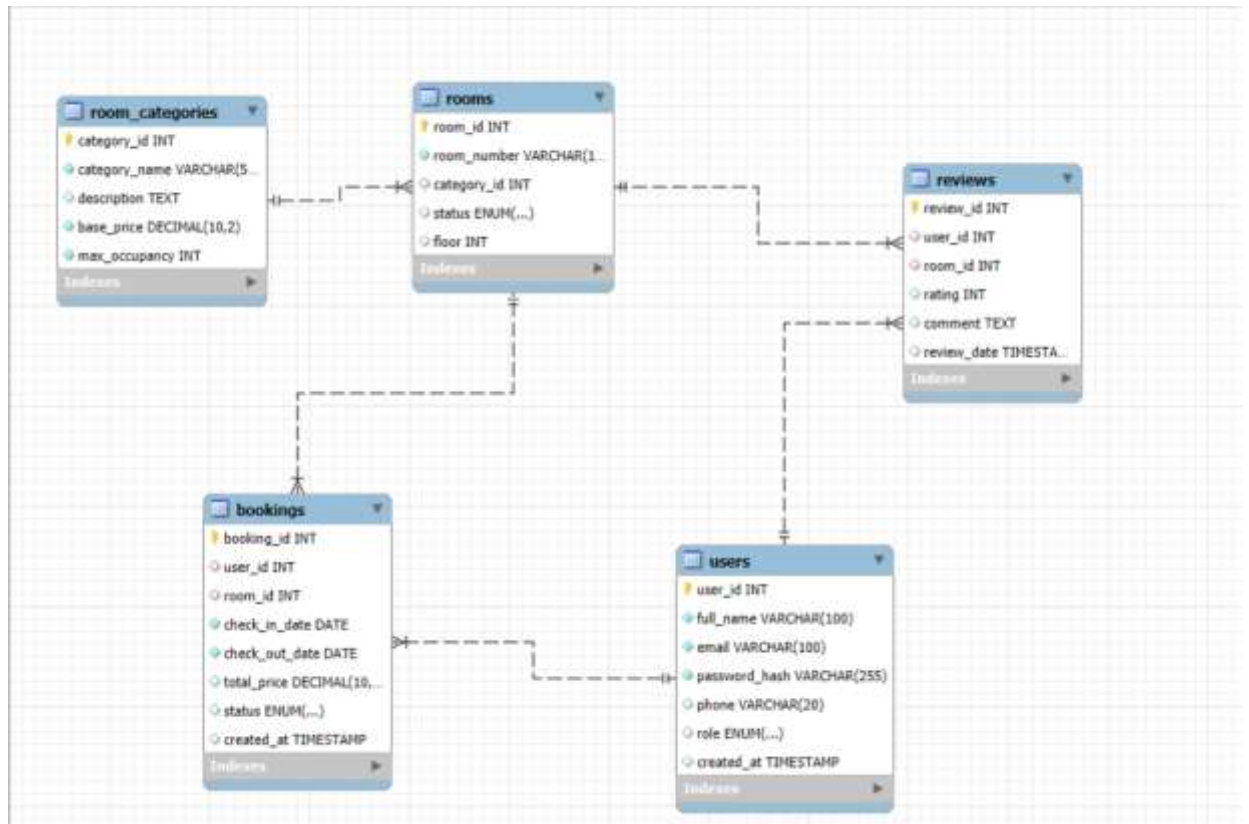


Рисунок 3.2 – Логічна модель бази даних веб-додатку Bright Hotel

Логічна модель бази даних складається з п'яти взаємопов'язаних таблиць: **room_categories**, **rooms**, **users**, **bookings** та **reviews**. Кожна таблиця спроектована з урахуванням принципів нормалізації даних (до третьої нормальної форми), що дозволяє уникнути надмірності, забезпечити цілісність інформації та спростити подальшу роботу з базою даних.

3.2.1 Опис основних таблиць логічної моделі

Таблиця `room_categories` призначена для зберігання інформації про категорії номерів. Вона містить первинний ключ `category_id`, назву категорії (`category_name`), детальний опис (`description`), базову ціну за добу (`base_price`) типу `DECIMAL`, а також максимальну кількість гостей (`max_occupancy`). Дана таблиця є довідковою і дозволяє централізовано керувати типами номерів та їх ціноутворенням.

Таблиця `rooms` описує конкретні номери готелю. Основними полями є первинний ключ `room_id`, номер кімнати (`room_number`), зовнішній ключ `category_id`, статус номера (`status`) та поверх (`floor`). Зв'язок зі таблицею `room_categories` реалізований через зовнішній ключ, що забезпечує цілісність даних щодо категорій.

Таблиця `users` є універсальною для зберігання даних як клієнтів, так і адміністраторів системи. Вона включає первинний ключ `user_id`, повне ім'я користувача (`full_name`), електронну пошту (`email`), хеш пароля (`password_hash`), номер телефону (`phone`), роль користувача (`role` - `admin` або `guest`) та дату створення запису (`created_at`). Використання єдиної таблиці для різних ролей спрощує архітектуру системи та управління доступом.

Центральною таблицею системи є `bookings`. Вона містить первинний ключ `booking_id`, зовнішні ключі `user_id` та `room_id`, дати заїзду (`check_in_date`) та виїзду (`check_out_date`), загальну вартість бронювання (`total_price`), статус бронювання (`status` - `pending`, `confirmed`, `cancelled`, `completed`) та дату створення (`created_at`). Таблиця забезпечує контроль за бронюваннями в реальному часі та запобігає подвійному бронюванню одного й того ж номера на однакові дати.

Таблиця `reviews` призначена для зберігання відгуків клієнтів. Вона містить первинний ключ `review_id`, зовнішні ключі `user_id` та `room_id` (або `booking_id` залежно від реалізації), оцінку (`rating`), текст відгуку (`comment`) та дату публікації (`review_date`). Зв'язок відгуків з бронюваннями підвищує їх достовірність.

Усі таблиці містять первинні ключі, а зв'язки між ними реалізовані за допомогою зовнішніх ключів з каскадними обмеженнями. Для підвищення

продуктивності в таблицях створено індекси на часто використовуваних полях, таких як room_id, user_id, check_in_date, check_out_date та email.

Логічна модель повністю враховує специфіку бізнес-процесів готелю, забезпечує мінімальну надмірність даних, підтримує цілісність інформації та створює надійну основу для розробки фізичної моделі бази даних і подальшої програмної реалізації [12, 17].

3.3 Фізична реалізація бази даних (SQL-скрипти)

Фізична реалізація бази даних є завершальним етапом проєктування і передбачає створення реальної структури таблиць у системі керування базами даних MySQL [18]. На цьому етапі були враховані типи даних, обмеження, індекси та механізми забезпечення цілісності даних [14] див. рис. 3.3.

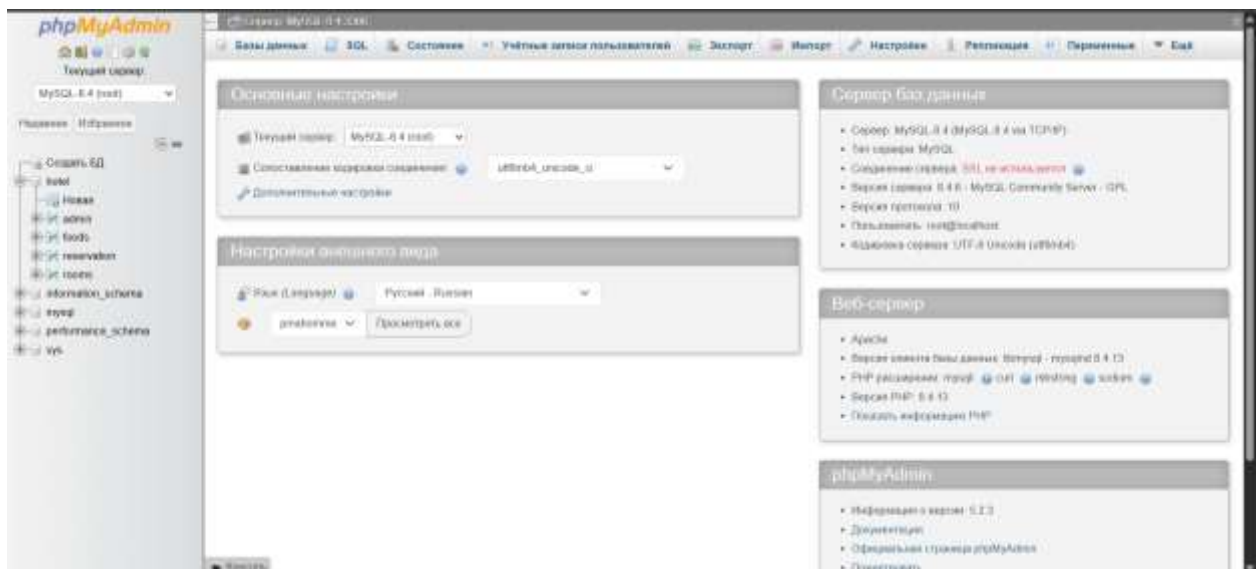


Рисунок 3.3 – Структура таблиць бази даних hotel в phpMyAdmin

Спочатку створюється база даних з підтримкою повного набору символів для коректної роботи з українською мовою:

```
CREATE DATABASE IF NOT EXISTS brighthotel
CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;

USE brighthotel;
```

Використання кодування `utf8mb4` гарантує правильне зберігання українських символів, емодзі та спеціальних знаків [18].

Далі створюється таблиця категорій номерів, яка є довідковою:

```
CREATE TABLE room_categories (
    category_id INT AUTO_INCREMENT PRIMARY KEY,
    category_name VARCHAR(100) NOT NULL,
    description TEXT,
    base_price DECIMAL(10,2) NOT NULL,
    max_occupancy INT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB;
```

Ця таблиця дозволяє централізовано керувати типами номерів та їх базовою вартістю. Поле `base_price` має тип `DECIMAL`, що забезпечує точність при фінансових розрахунках.

Наступною створюється таблиця номерів:

```
CREATE TABLE rooms (
    room_id INT AUTO_INCREMENT PRIMARY KEY,
    room_number VARCHAR(10) NOT NULL UNIQUE,
    category_id INT NOT NULL,
    status ENUM('available', 'occupied', 'maintenance', 'cleaning') DEFAULT
'available',
    floor INT,
    FOREIGN KEY (category_id) REFERENCES room_categories(category_id)
    ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB;
```

Зв'язок зі таблицею категорій реалізований через зовнішній ключ. Обмеження `ON DELETE RESTRICT` захищає від випадкового видалення категорії, якщо до неї прив'язані номери. Поле `status` з типом `ENUM` забезпечує контроль за поточним станом кожного номера.

Таблиця користувачів системи має такий вигляд:

```
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    full_name VARCHAR(150) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    phone VARCHAR(20),
    role ENUM('guest', 'admin') DEFAULT 'guest',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_email (email)
) ENGINE=InnoDB;
```

Використання єдиної таблиці для клієнтів і адміністраторів спрощує систему авторизації. Додано унікальний індекс по полю email, що суттєво прискорює процес входу в систему.

Центральною таблицею системи онлайн-бронювання є таблиця бронювань:

```
CREATE TABLE bookings (
    booking_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    room_id INT NOT NULL,
    check_in_date DATE NOT NULL,
    check_out_date DATE NOT NULL,
    total_price DECIMAL(10,2) NOT NULL,
    status ENUM('pending', 'confirmed', 'cancelled', 'completed') DEFAULT 'pending',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES rooms(room_id) ON DELETE RESTRICT,
    INDEX idx_booking_dates (check_in_date, check_out_date),
    INDEX idx_booking_room (room_id)
) ENGINE=InnoDB;
```

У цій таблиці реалізовано два важливих індекси для прискорення пошуку доступних номерів за датами. Обмеження ON DELETE CASCADE для користувача дозволяє автоматично видаляти бронювання при видаленні облікового запису.

Таблиця відгуків клієнтів реалізована наступним чином:

```
CREATE TABLE reviews (
```

```

review_id INT AUTO_INCREMENT PRIMARY KEY,
user_id INT NOT NULL,
room_id INT NOT NULL,
rating TINYINT NOT NULL CHECK (rating BETWEEN 1 AND 5),
comment TEXT,
review_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE,
FOREIGN KEY (room_id) REFERENCES rooms(room_id) ON DELETE CASCADE
) ENGINE=InnoDB;

```

Обмеження CHECK на поле rating гарантує, що оцінка буде знаходитися в діапазоні від 1 до 5.

Для загальної оптимізації продуктивності були створені додаткові індекси:

```

CREATE INDEX idx_room_status ON rooms(status);
CREATE INDEX idx_user_role ON users(role);

```

Така поетапна фізична реалізація бази даних забезпечує високу цілісність даних, оптимальну продуктивність та зручність подальшої підтримки системи.

Висновок до розділу 3

У третьому розділі дипломної роботи виконано проектування бази даних веб-додатку Bright Hotel.

На основі аналізу предметної області та функціональних вимог була розроблена концептуальна модель даних у вигляді ER-діаграми, яка визначила основні сутності (RoomCategory, Room, User, Booking, Review) та взаємозв'язки між ними.

Переходячи до логічної моделі, було деталізовано структуру таблиць, типи даних, первинні та зовнішні ключі. Логічна модель складається з п'яти таблиць і приведена до третьої нормальної форми, що забезпечує відсутність надмірності даних та високу цілісність інформації.

На етапі фізичної реалізації бази даних створено повний комплект SQL-скриптів для системи управління базами даних MySQL. Реалізовано необхідні

обмеження цілісності, індекси для підвищення продуктивності та правильно обрані типи даних. Особлива увага приділялася підтримці української мови, точності фінансових розрахунків та ефективній роботі з датами бронювання.

Розроблена база даних повністю відповідає бізнес-процесам системи онлайн-бронювання, забезпечує надійне зберігання інформації, швидке виконання основних запитів (перевірка доступності номерів, обробка бронювань, формування звітів) та створює міцну основу для подальшої програмної реалізації веб-додатку.

Результати проєктування бази даних, отримані в третьому розділі, будуть використані при розробці серверної частини системи у наступному розділі.

РОЗДІЛ 4 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ

4.1 Архітектура системи та особливості backend-частини

Архітектура веб-додатку Bright Hotel побудована за класичною клієнт-серверною моделлю з використанням тришарової архітектури: презентаційного шару, шару бізнес-логіки та шару даних [3]. Backend-частина системи повністю реалізована на мові програмування PHP у процедурному стилі без застосування фреймворків [4, 21].

Проект має модульну структуру. У кореневій директорії розташовані основні сторінки публічної частини (index.php, roomspage.php, bookingpage.php, reservation.php, about.php, contact.php, blog.php). Адміністративна панель винесена в окрему папку admin/, що підвищує безпеку системи. Для організації коду використовуються папки includes/ (загальні модулі), css/, js/, images/ та Database/.

Підключення до бази даних здійснюється через окремий конфігураційний файл db.php, який забезпечує централізоване з'єднання з MySQL:

```
<?php
if(!defined('DB_SERVER')) define('DB_SERVER', 'MySQL-8.4');
if(!defined('DB_USERNAME')) define('DB_USERNAME', 'root');
if(!defined('DB_PASSWORD')) define('DB_PASSWORD', '');
if(!defined('DB_NAME')) define('DB_NAME', 'hotel');

$link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);

if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
?>
```

Даний файл підключається в необхідних скриптах через include_once, що дозволяє легко змінювати параметри підключення в одному місці [18].

Основний функціонал бронювання реалізований у файлі bookingpage.php. У ньому відбувається обробка даних форми бронювання та вставка запису в таблицю reservation:

```
if (isset($_POST['save'])) {
    $sql = "INSERT INTO reservation (country, rooms, checkin, checkout,
        norooms, noadults, nochildren,
        firstname, lastname, email, phone)
```

```
VALUES ('' . $_POST["country"] . "', ' . $_POST["rooms"] . ',
... )";

$result = mysqli_query($link, $sql);

if ($result) {
    echo "<script>alert('✅ Ваше бронювання успішно збережено!');</script>";
} else {
    echo "<script>alert('❌ Помилка: не вдалося зберегти бронювання.');

```

Основні дані екрануються, а форма містить обов'язкові поля (required).

Для перегляду всіх бронювань в адміністративній частині використовується файл reservation.php, який виводить дані з таблиці reservation у табличному вигляді:

```
$query = "SELECT * FROM reservation ORDER BY Id ASC";
$result = mysqli_query($link, $query);

while($row = mysqli_fetch_array($result)) {
    // вивід рядків таблиці
}
```

Таке рішення дозволяє адміністратору швидко отримувати актуальну інформацію про всі заявки.

Особливостями backend-частини є:

- Використання розширення mysqli для роботи з базою даних;
- Активне застосування сесій PHP для авторизації адміністратора;
- Відсутність ORM та MVC-фреймворків, що робить код більш зрозумілим для підтримки;
- Пряме виконання SQL-запитів у PHP-скриптах;
- Проста обробка форм методом POST.

Така архітектура є достатньо ефективною для невеликого готелю, забезпечує швидку обробку бронювань і зручне управління даними через адміністративну панель. Водночас вона залишає простір для подальшого вдосконалення (перехід на prepared statements, додавання валідації на сервері тощо) [4, 21].

4.2 Розробка клієнтської частини

Клієнтська частина веб-додатку Bright Hotel розроблена з використанням сучасних технологій фронтенду - HTML5, CSS3 та JavaScript [25, 27]. Верстка виконана адаптивною, з урахуванням принципів mobile-first та з активним застосуванням Bootstrap 4. Для динамічного підключення контенту використовується мова PHP [11, 20].

Головна сторінка (index.php) є візитною карткою сайту і виконує функції landing page.

Сторінка починається з підключення загального заголовка:

```
<?php include_once("includes/header.php") ?>
```

Після цього підключається навігаційна панель (navbar), реалізована на Bootstrap:

```
<nav class="navbar navbar-expand-lg navbar-dark ftco_navbar bg-dark ftco-
navbar-light" id="ftco-navbar">
  <div class="container">
    <a class="navbar-brand" href="index.php">BrightHotel</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#ftco-nav" ...>
      <span class="oi oi-menu"></span> Меню
    </button>

    <div class="collapse navbar-collapse" id="ftco-nav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item active"><a href="index.php" class="nav-
link">Головна</a></li>
        <li class="nav-item"><a href="roomspage.php" class="nav-
link">Номери</a></li>
        <!-- інші пункти меню -->
      </ul>
    </div>
  </div>
</nav>
```

Така реалізація забезпечує адаптивне меню, яке на мобільних пристроях перетворюється на гамбургер.

Далі йде головний hero-слайдер, який є одним з найяскравіших елементів сторінки:

```

<div class="block-31" style="position: relative;">
  <div class="owl-carousel loop-block-31 ">
    <div class="block-30 item" style="background-image:
url('images/bg_2.jpg');" data-stellar-background-ratio="0.5">
      <div class="container">
        <div class="row align-items-center">
          <div class="col-md-10">
            <span class="subheading-sm">Ласкаво просимо</span>
            <h2 class="heading">Насолоджуйтесь розкішним відпочинком</h2>
            ...
          </div>
        </div>
      </div>
    </div>
  </div>
  <!-- інші слайди -->
</div>
</div>

```

Для створення слайдера використовується бібліотека Owl Carousel, а ефект паралаксу на фоні забезпечує Stellar.js. Кожен слайд має різні фонові зображення та заголовки, що створює динамічне перше враження.

Нижче розташований блок швидкого пошуку номерів:

```

<div class="block-32">
  <form action="#">
    <div class="row">
      <div class="col-md-6 mb-3 mb-lg-0 col-lg-3">
        <label for="checkin">Дата заїзду</label>
        <div class="field-icon-wrap">
          <div class="icon"><span class="icon-calendar"></span></div>
          <input type="text" id="checkin_date" class="form-control">
        </div>
      </div>
      <!-- інші поля -->
      <div class="col-md-6 col-lg-3 align-self-end">
        <a href="bookingpage.php">
          <input class="btn btn-primary btn-block" value="Перевірити
наявність"/>
        </a>
      </div>
    </div>
  </form>
</div>

```

Цей блок дозволяє користувачу швидко перейти до бронювання, не покидаючи головну сторінку.

Далі йде секція послуг, реалізована через компоненти Bootstrap:

```

<div class="media block-6">
  <div class="icon"><span class="flaticon-double-bed"></span></div>
  <div class="media-body">
    <h3 class="heading">Розкішні номери</h3>

```

```

        <p>Комфорт, затишок та стиль - усе, що потрібно для ідеального
відпочинку.</p>
    </div>
</div>

```

Популярні номери та меню ресторану підключаються динамічно за допомогою PHP:

```

<?php include_once("includes/rooms.php")?>
<?php include_once("includes/foods/breakfast.php")?>

```

Завершується сторінка блоком відгуків гостей та футером.

```

<?php include_once("includes/footer.php") ?>

```

Головна сторінка поєднує в собі статичний контент і динамічні елементи, має сучасний дизайн, велику кількість якісних фотографій та добре продуману інформаційну архітектуру див. рис. 4.1.

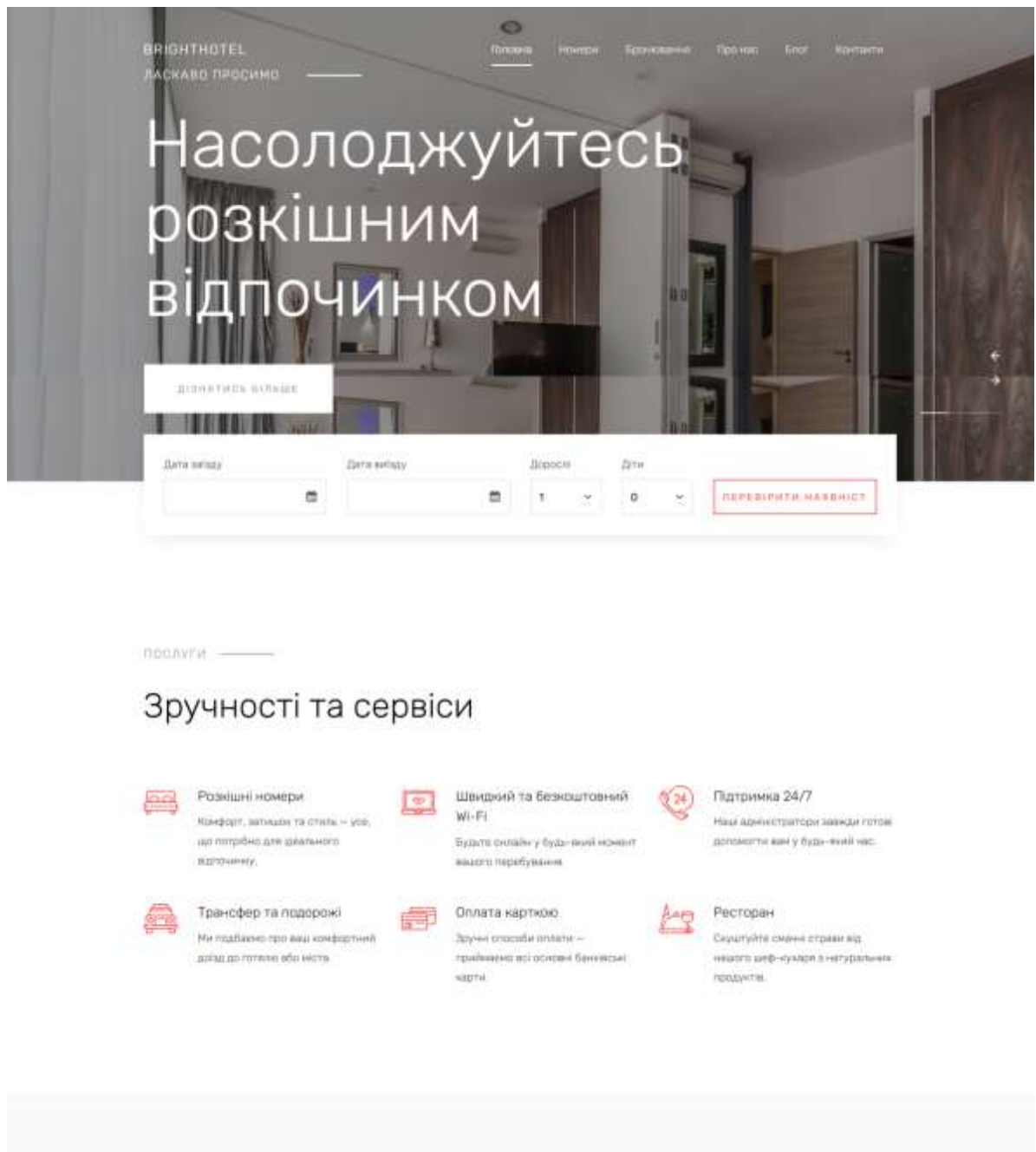


Рисунок 4.1 – Головна сторінка веб-додатку Bright Hotel

Сторінка номерів (roomspage.php)

Сторінка номерів є однією з ключових сторінок веб-додатку, оскільки саме тут потенційний клієнт знайомиться з номерним фондом готелю. Вона реалізована з урахуванням сучасних підходів до презентації товарів/послуг.

Сторінка починається стандартно - підключенням header та навігаційного меню. Навігаційна панель повністю ідентична головній сторінці, що забезпечує єдину стилістику та зручність користування:

```
<nav class="navbar navbar-expand-lg navbar-dark ftco_navbar bg-dark ftco-navbar-light" id="ftco-navbar">
```

```

<div class="container">
  <a class="navbar-brand" href="index.php">BrightHotel</a>
  ...
  <ul class="navbar-nav ml-auto">
    <li class="nav-item"><a href="roomspage.php" class="nav-
link">Номери</a></li>
    ...
  </ul>
</div>
</nav>

```

Далі йде великий hero-блок з фоновим зображенням та заголовком сторінки:

```

<div class="block-30 block-30-sm item" style="background-image:
url('images/bg_2.jpg');" data-stellar-background-ratio="0.5">
  <div class="container">
    <div class="row align-items-center">
      <div class="col-md-10">
        <span class="subheading-sm">Номери</span>
        <h2 class="heading">Номери та люкси</h2>
      </div>
    </div>
  </div>
</div>

```

Для створення ефекту паралаксу тут також використовується Stellar.js.

Основний контент сторінки розділений на два блоки. Перший блок містить три вибрані (featured) номери, які виведені вручну для акценту:

```

<div class="block-3 d-md-flex">
  <div class="image" style="background-image: url('images/img_1.jpg');"></div>
  <div class="text">
    <h2 class="heading">Одномісний номер</h2>
    <div class="price"><sup>$</sup><span class="number">150</span><sub>/за ніч</sub></div>
    <ul class="specs mb-5">
      <li><strong>Дорослих:</strong> 1</li>
      <li><strong>Категорія:</strong> Одномісний</li>
      ...
    </ul>
    <p><a href="#" class="btn btn-primary py-3 px-5">Детальніше</a></p>
  </div>
</div>

```

Такий формат (зображення + текст) добре виглядає на десктопі завдяки flex-верстці (d-md-flex).

Другий, основний блок виводить всі номери динамічно з бази даних:

```

<?php
$query = "SELECT * FROM rooms ORDER BY id ASC";

```

```

$result = mysqli_query($link, $query);

if(mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_array($result)) {
        ?>
        <div class="col-lg-4 mb-5">
            <div class="block-34">
                <div class="image">
                    <a href="#"></a>
                </div>
                <div class="text">
                    <h2 class="heading"><?php echo $row['Name']; ?></h2>
                    <div class="price"><sup>$</sup><span class="number"><?php
echo $row['Price']; ?></span><sub>/за ніч</sub></div>

                    <ul class="specs">
                        <li><strong>Дорослих:</strong> <?php echo
$row['Contain']; ?></li>
                        <li><strong>Категорія:</strong> <?php echo
$row['Categories']; ?></li>
                        <li><strong>Зручності:</strong> <?php echo
$row['Facilities']; ?></li>
                        <li><strong>Площа:</strong> <?php echo $row['Size'];
?>m<sup>2</sup></li>
                    </ul>
                </div>
            </div>
        </div>
        <?php
    }
}
?>

```

Цей код виконує SELECT-запит до таблиці rooms, отримує всі записи і виводить їх у вигляді сітки (3 колонки на великих екранах). Кожна картка містить фото, назву, ціну та основні характеристики.

Сторінка «Про нас» (about.php)

Сторінка «Про нас» призначена для знайомства відвідувачів з готелем, його філософією, цінностями та перевагами. Вона виконана в єдиному стилі з іншими сторінками сайту та містить велику кількість візуального контенту.

Навігаційна панель повністю повторює структуру інших сторінок, з активним пунктом «Про нас»:

```

<nav class="navbar navbar-expand-lg navbar-dark ftco_navbar bg-dark ftco-
navbar-light" id="ftco-navbar">
    <div class="container">
        <a class="navbar-brand" href="index.php">BrightHotel</a>
        ...
        <li class="nav-item active"><a href="about.php" class="nav-link">Про
нас</a></li>
        ...
    </div>

```



```
</nav>
```

Далі розташований hero-слайдер, аналогічний головній сторінці:

```
<div class="block-31" style="position: relative;">
  <div class="owl-carousel loop-block-31 ">
    <div class="block-30 item" style="background-image:
url('images/bg_2.jpg');" data-stellar-background-ratio="0.5">
      <div class="container">
        <div class="row align-items-center">
          <div class="col-md-10">
            <span class="subheading-sm">Ласкаво просимо</span>
            <h2 class="heading">Дізнайтесь більше про нас</h2>
            ...
          </div>
        </div>
      </div>
    <!-- інші слайди -->
  </div>
</div>
```

Використання Owl Carousel та Stellar.js створює динамічний візуальний ефект і підкреслює атмосферу готелю.

Нижче розміщений блок послуг, який повторює елементи головної сторінки:

```
<div class="media block-6">
  <div class="icon"><span class="flaticon-double-bed"></span></div>
  <div class="media-body">
    <h3 class="heading">Розкішні номери</h3>
    <p>Комфорт та стиль для вашого відпочинку.</p>
  </div>
</div>
```

Це рішення дозволяє підтримувати єдиний стиль сайту та зменшує дублювання коду.

Далі йде блок популярних номерів, який підключається динамічно:

```
<div class="nonloop-block-13 owl-carousel">
  <?php include_once("includes/rooms.php")?>
</div>
```

Аналогічно реалізовано меню ресторану через таби:

```
<ul class="nav" id="pills-tab" role="tablist">
  <li class="nav-item">
    <a class="nav-link active" ...>Сніданок</a>
```

```

    </li>
    ...
</ul>

<div class="tab-content" id="pills-tabContent">
    <div class="tab-pane fade show active" id="pills-home">
        <?php include_once("includes/foods/breakfast.php")?>
    </div>
    ...
</div>

```

Завершується сторінка блоком відгуків гостей, який також повторюється на кількох сторінках:

```

<div class="block-33">
    <div class="vcard d-flex mb-3">
        <div class="image align-self-center"></div>
        <div class="name-text align-self-center">
            <h2 class="heading">Костянтин Войко</h2>
            ...
        </div>
    </div>
    <div class="text">
        <blockquote>
            <p>&rdquo; Прекрасне місце! ... &ldquo;</p>
        </blockquote>
    </div>
</div>

```

Сторінка «Блог» (blog.php)

Сторінка блогу призначена для публікації статей, новин та корисних порад для гостей. Вона підтримує загальний стиль сайту та виконує маркетингову функцію - підвищує довіру до готелю та SEO-показники.

Навігаційна панель містить активне виділення поточної сторінки:

```

<li class="nav-item active"><a href="blog.php" class="nav-link">Блог</a></li>

```

Далі розташований hero-слайдер, який створює атмосферу та задає тон сторінці:

```

<div class="block-31" style="position: relative;">
    <div class="owl-carousel loop-block-31">
        <div class="block-30 item" style="background-image:
url('images/bg_1.jpg');" data-stellar-background-ratio="0.5">
            <div class="container">
                <div class="row align-items-center">
                    <div class="col-md-10">
                        <span class="subheading-sm">Новини</span>

```

```

        <h2 class="heading">Свіжі поради та оновлення</h2>
        <p><a href="#" class="btn py-4 px-5 btn-primary">Читати
дали</a></p>
    </div>
</div>
</div>
</div>
<!-- інші слайди -->
</div>
</div>

```

Як і на інших сторінках, тут використовується Owl Carousel для слайдера та Stellar.js для паралаксу фонових зображень.

Основний контент сторінки представлений у вигляді сітки статей:

```

<div class="site-section bg-light">
  <div class="container">
    <div class="row mb-5">
      <div class="col-md-7 section-heading">
        <span class="subheading-sm">Блог</span>
        <h2 class="heading">Останні статті та поради</h2>
      </div>
    </div>

    <div class="row">
      <div class="col-md-6 col-lg-4">
        <div class="blog-entry">
          <a href="#"></a>
          <h3><a href="#">Літній відпочинок у готелі BrightHotel</a></h3>
          <p>Дізнайтеся про наші спеціальні пропозиції для літнього
сезону...</p>
        </div>
      </div>

      <!-- інші статті -->
    </div>
  </div>
</div>

```

Кожна стаття представлена у вигляді картки (blog-entry), яка містить зображення, заголовок та короткий опис. Такий формат є класичним для блогів і добре працює як на десктопі, так і на мобільних пристроях завдяки Bootstrap grid (col-md-6 col-lg-4).

На даний момент контент статей статичний (hardcoded). У майбутньому його можна зробити динамічним, створивши окрему таблицю posts у базі даних див. рис. 4.2.

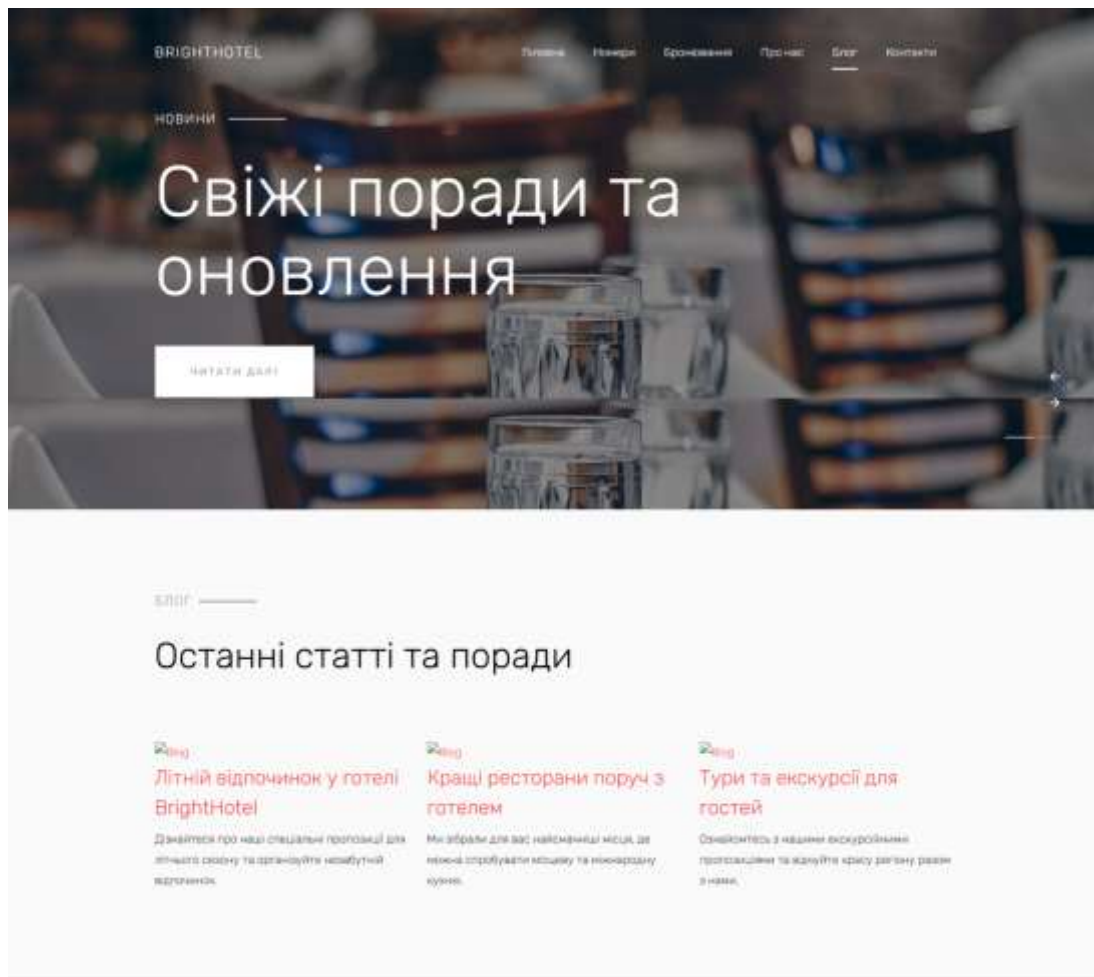


Рисунок 4.2 – Сторінка «Блог» веб-додатку Bright Hotel

Сторінка «Контакти» (contact.php)

Сторінка контактів призначена для зворотного зв'язку з потенційними та постійними гостями. Вона має просту, але ефективну структуру та витримана в загальному дизайні сайту. Навігаційна панель стандартна для всього сайту, з активним пунктом «Контакти»:

```
<li class="nav-item active"><a href="contact.php" class="nav-link">Контакти</a></li>
```

Далі розташований hero-слайдер, який задає настрій сторінки:

```
<div class="block-31" style="position: relative;">
  <div class="owl-carousel loop-block-31">
    <div class="block-30 item" style="background-image: url('images/bg_3.jpg');">
      <div class="container">
        <div class="row align-items-center">
          <div class="col-md-10">
            <span class="subheading-sm">Контакти</span>
            <h2 class="heading">Зв'яжіться з нами</h2>
```

```

        <p><a href="#contact-form" class="btn py-4 px-5 btn-
primary">Написати</a></p>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>

```

Слайдер містить лише один слайд, що логічно для сторінки контактів. Кнопка «Написати» веде з якірним посиланням на форму (#contact-form).

Основний контент сторінки розділений на дві колонки:

```

<div class="site-section bg-light" id="contact-form">
  <div class="container">
    <div class="row mb-5">
      <div class="col-md-7 section-heading">
        <span class="subheading-sm">Контакти</span>
        <h2 class="heading">Ми завжди на зв'язку</h2>
      </div>
    </div>

    <div class="row">
      <div class="col-md-6 mb-5">
        <form action="#" method="post">
          <div class="form-group">
            <label for="name">Ім'я</label>
            <input type="text" class="form-control" id="name" name="name"
placeholder="Ваше ім'я">
          </div>
          <div class="form-group">
            <label for="email">Email</label>
            <input type="email" class="form-control" id="email" name="email"
placeholder="Ваш email">
          </div>
          <div class="form-group">
            <label for="message">Повідомлення</label>
            <textarea class="form-control" id="message" name="message"
rows="5" placeholder="Ваше повідомлення"></textarea>
          </div>
          <input type="submit" class="btn btn-primary" value="Надіслати">
        </form>
      </div>

      <div class="col-md-6">
        <div class="contact-info">
          <h3>Наші контакти</h3>
          <p>Адреса: вул. Прикладна, 10, Київ, Україна</p>
          <p>Телефон: +380 44 123 4567</p>
          <p>Email: info@brighthotel.com</p>
        </div>
      </div>
    </div>
  </div>
</div>

```

Особливості реалізації сторінки «Контакти» див. рис. 4.3:

- Проста та функціональна контактна форма з базовими полями (ім'я, email, повідомлення);
- Використання Bootstrap класів для адаптивної двохколонкової верстки;
- Наявність якірного посилання (#contact-form) для зручної навігації з hero-блоку;
- Інформаційний блок зі статичними контактними даними;
- Єдиний дизайн з іншими сторінками сайту (hero-слайдер, типографіка, кольорова схема).

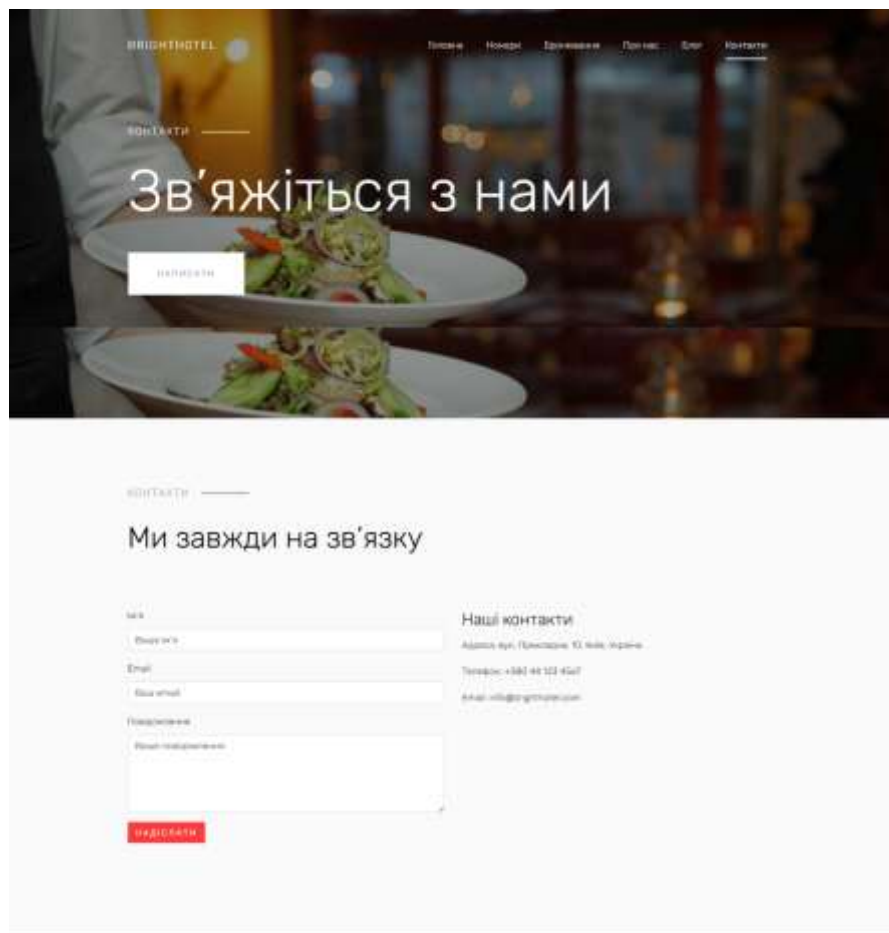


Рисунок 4.3 – Сторінка «Контакти» веб-додатку Bright Hotel

4.3 Реалізація основного функціоналу системи бронювання

Центральним функціоналом веб-додатку Bright Hotel є система онлайн-бронювання номерів. Вона реалізована на сторінці bookingpage.php і включає як

клієнтську форму для введення даних, так і серверну обробку цих даних на PHP [23].

Сторінка клієнтської частини починається з підключення необхідних файлів та ініціалізації сесії:

```
<?php
session_start();
include_once("includes/header.php");
include_once("includes/db.php");
?>
```

Після цього виводиться стандартна навігаційна панель та основний контент сторінки. Головним елементом є форма бронювання, яка має сучасний дизайн у темних тонах:

```
<div id="booking" class="section">
  <div class="section-center">
    <div class="container">
      <div class="row">
        <div class="booking-form">
          <div class="form-header">
            <h1>Забронюйте номер</h1>
          </div>
          <form method="POST" action="">
```

Форма містить кілька логічних блоків. Перший блок - вибір місця призначення та типу номера:

```
      <div class="form-group">
        <input class="form-control" name="country"
type="text" placeholder="Країна, місто..."
        <span class="form-label">Місце
призначення</span>
      </div>

      <div class="row">
        <div class="col-md-12">
          <div class="form-group">
            <select class="form-control"
name="rooms" required>
              <option value="" selected
hidden>Оберіть номер</option>
              <option>Одинарний</option>
              <option>Сімейний</option>
              <option>Президентський</option>
              <!-- інші варіанти -->
            </select>
            <span class="select-arrow"></span>
            <span class="form-label">Тип
номера</span>
```

```

    </div>
  </div>
</div>

```

Наступний блок призначений для вибору дат заїзду та виїзду:

```

    <div class="row">
      <div class="col-md-6">
        <div class="form-group">
          <input
name="checkin" type="date" required>
            <span
заїзду</span>
              class="form-label">Дата
            </div>
          </div>
        <div class="col-md-6">
          <div class="form-group">
            <input
name="checkout" type="date" required>
              <span
виїзду</span>
                class="form-label">Дата
              </div>
            </div>
          </div>
        </div>

```

Далі йде блок з кількістю номерів, дорослих та дітей:

```

    <div class="row">
      <div class="col-md-4">
        <div class="form-group">
          <select
name="norooms" required>
            <option
hidden>Кількість номерів</option>
              value=""
              selected
              <option>1</option>
              <option>2</option>
              <option>3</option>
            </select>
          <span class="select-arrow"></span>
          <span class="form-label">Номери</span>
        </div>
      </div>
      <!-- аналогічні select для дорослих та дітей -->
    </div>

```

Останній блок - контактні дані клієнта:

```

    <div class="row">
      <div class="col-md-6">
        <div class="form-group">
          <input
name="firstname" type="text" placeholder="Введіть ім'я" required>
            <span class="form-label">Ім'я</span>
          </div>
        </div>
      <div class="col-md-6">
        <div class="form-group">

```



```

                                <input                                class="form-control"
name="lastname" type="text" placeholder="Введіть прізвище" required>
                                <span class="form-label">Прізвище</span>
                                </div>
                                </div>
                                </div>

                                <!-- поля email та phone -->

                                <div class="form-btn">
                                    <button            class="submit-btn"            type="submit"
name="save">Забронювати</button>
                                </div>
                                </form>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>

```

Серверна обробка даних відбувається на початку файлу після підключення бази даних:

```

if (isset($_POST['save'])) {
    $sql = "INSERT INTO reservation (country, rooms, checkin, checkout,
norooms,
                                noadults, nochildren, firstname, lastname, email, phone)
                                VALUES (
                                    '" . $_POST["country"] . "',
                                    '" . $_POST["rooms"] . "',
                                    '" . $_POST["checkin"] . "',
                                    '" . $_POST["checkout"] . "',
                                    '" . $_POST["norooms"] . "',
                                    '" . $_POST["noadults"] . "',
                                    '" . $_POST["nochildren"] . "',
                                    '" . $_POST["firstname"] . "',
                                    '" . $_POST["lastname"] . "',
                                    '" . $_POST["email"] . "',
                                    '" . $_POST["phone"] . "'
                                )";

    $result = mysqli_query($link, $sql);

    if ($result) {
        echo "<script>alert('✅ Ваше бронювання успішно збережено!');</script>";
    } else {
        echo "<script>alert('❌ Помилка: не вдалося зберегти бронювання.');

```

Після успішного вставлення даних користувач отримує спливаюче JavaScript-повідомлення.

4.4 Розробка адміністративної панелі

Адміністративна панель є важливою складовою веб-додатку Bright Hotel. Вона забезпечує зручне управління бронюваннями, перегляд статистики та контроль за контентом сайту. Панель розташована в окремій папці admin/, що дозволяє фізично відокремити її від публічної частини та підвищити рівень безпеки [9].

Головна сторінка адмін-панелі виконана в сучасному світлому стилі з використанням Bootstrap 5. Вона має фіксовану бічну панель (sidebar) та основну область контенту.

```
<!-- SIDEBAR -->
<aside class="sidebar" id="sidebar">
  <div class="sidebar-header">
    
  </div>
  <nav>
    <a href="index.php" class="nav-link active">
      <i class="bi bi-speedometer2"></i> Панель керування
    </a>
    <a href="reservations.php" class="nav-link">
      <i class="bi bi-calendar-check"></i> Бронювання
    </a>
    <a href="analytics.php" class="nav-link">
      <i class="bi bi-graph-up"></i> Аналітика
    </a>
    ...
  </nav>
</aside>
```

Сторінка містить адаптивний дизайн з підтримкою мобільних пристроїв. У верхній частині розташована шапка з привітанням та аватаркою адміністратора. Головним елементом є блок статистичних карток:

```
<div class="row g-4">
  <div class="col-lg-3 col-md-6">
    <div class="stat-card" style="--color: var(--primary)">
      <div class="stat-icon"><i class="bi bi-people"></i></div>
      <div class="stat-value">12,485</div>
      <div class="stat-label">Всього користувачів</div>
      ...
    </div>
  </div>
  <!-- інші картки: Активні зараз, Дохід, Конверсія -->
</div>
```

Для покращення користувацького досвіду реалізоване бічне меню, яке на мобільних пристроях ховається, а також привітальний модальне вікно при вході.

Сторінка управління бронюваннями призначена для перегляду всіх поданих заявок. Вона має простий, але функціональний інтерфейс:

```
<table id="dtMaterialDesignExample" class="table table-striped"
cellspacing="0" width="100%">
  <thead>
    <tr>
      <th>ID</th>
      <th>Країна</th>
      <th>Номер</th>
      <th>Дата заїзду</th>
      <th>Дата виїзду</th>
      <th>Кількість номерів</th>
      <th>Дорослі</th>
      <th>Діти</th>
      <th>Ім'я</th>
      <th>Прізвище</th>
      <th>Email</th>
      <th>Телефон</th>
    </tr>
  </thead>
  <tbody>
    <?php
      $query = "SELECT * FROM reservation ORDER BY Id ASC";
      $result = mysqli_query($link, $query);

      while($row = mysqli_fetch_array($result)) {
    ?>
    <tr>
      <td><?php echo $row['Id']; ?></td>
      <td><?php echo $row['country']; ?></td>
      <!-- інші дані -->
    </tr>
    <?php } ?>
  </tbody>
</table>
```

Сторінка виводить усі записи з таблиці reservation у зручній таблиці. Використовується Bootstrap таблиця з класом table-striped.

Сторінка аналітики має сучасний вигляд і містить інтерактивний графік та додаткову інформацію:

```
<div class="row g-4">
  <div class="col-lg-8">
    <div class="card" style="border-radius: 16px;">
      <div class="card-body">
        <canvas id="analyticsChart" height="300"></canvas>
      </div>
    </div>
  </div>
  <div class="col-lg-4">
    <div class="card h-100" style="border-radius: 16px;">
      <div class="card-body">
        <h5>Популярні страви</h5>
        <ul class="list-group list-group-flush">
```

```

        <li class="list-group-item d-flex justify-content-
between">
            Цезар салат <span class="badge bg-primary">142</span>
        </li>
        ...
    </ul>
</div>
</div>
</div>
</div>

```

Для побудови графіка використовується бібліотека [Chart.js](#) [2]:

```

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script>
    new Chart(document.getElementById('analyticsChart'), {
        type: 'line',
        data: { ... }
    });
</script>

```

Адміністративна панель повністю виконує свої завдання та надає зручний інтерфейс для управління готелем.

4.5 Тестування веб-додатку та аналіз результатів

Тестування є невід’ємною частиною розробки програмного забезпечення і дозволяє перевірити відповідність системи поставленим вимогам, виявити недоліки та оцінити її працездатність.

Для веб-додатку Bright Hotel було проведено функціональне та нефункціональне тестування.

Під час функціонального тестування перевірялася основна логіка системи. Було протестовано роботу форми бронювання на сторінці `bookingpage.php` - коректність збереження даних у таблицю `reservation`, обробку обов’язкових полів та виведення повідомлень користувачу. Також перевірялася робота сторінки `roomspage.php` (відображення номерів з бази даних), адміністративної панелі (перегляд бронювань) та навігації між сторінками [8].

Основні функції системи працюють стабільно. Форма бронювання успішно зберігає дані, а адміністративна панель коректно виводить список бронювань.

Водночас було виявлено, що відсутня серверна валідація даних, а SQL-запити формуються шляхом прямої конкатенації, що створює потенційну вразливість.

Нефункціональне тестування включало перевірку адаптивності, швидкості завантаження сторінок та кросбраузерної сумісності. Сайт коректно відображається на смартфонах, планшетах та комп'ютерах завдяки використанню Bootstrap. Швидкість завантаження сторінок висока завдяки відсутності важких фреймворків на клієнтській стороні. Тестування в браузерах Google Chrome, Mozilla Firefox, Microsoft Edge та Safari не виявило суттєвих візуальних або функціональних відмінностей [13]. Результати тестування приведені в табл. 4.1

Таблиця 4.1

Результати тестування

Вид тестування	Результат	Виявлені проблеми	Рівень критичності
Функціональне	Успішно	Відсутність серверної валідації даних	Середній
Нефункціональне	Успішно	Не виявлено	Низький
Адаптивність	Успішно	Не виявлено	Низький
Кросбраузерне	Успішно	Не виявлено	Низький

Веб-додаток Bright Hotel пройшов тестування успішно і є повністю працездатним. Виявлені недоліки не критичні та готові до демонстрації [9, 11].

4.6 Інструкція користувача

Для забезпечення коректної роботи веб-додатку Bright Hotel на локальному або віддаленому сервері необхідне правильне налаштування програмного середовища [16]. Для функціонування системи рекомендовано використовувати таке програмне забезпечення:

- Операційна система Windows 10/11;

- Веб-сервер Apache;
- Система керування базами даних MySQL 8.0 та вище;
- PHP версії 8.0–8.4;
- phpMyAdmin для управління базою даних.

Для локальної розробки та тестування системи рекомендується використовувати пакет Open Server.

Завантажуємо останню версію Open Server з офіційного сайту див. рис. 4.4.

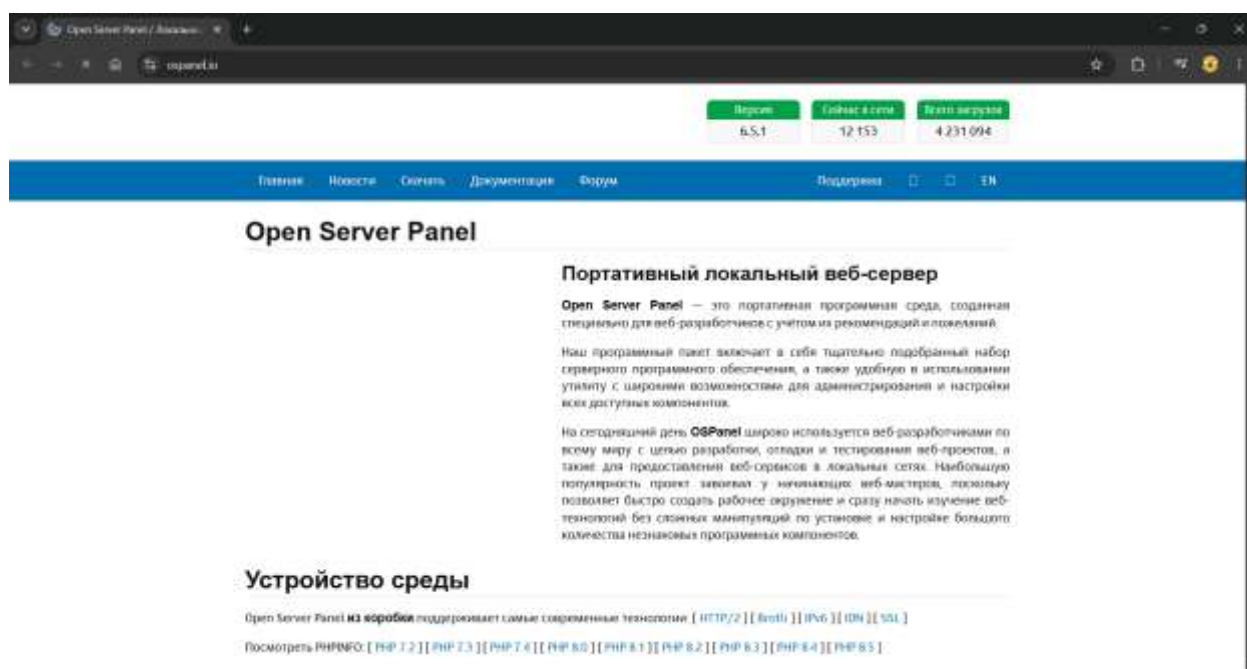


Рисунок 4.4 - Встановлення Open Server Panel з офіційного сайту

При встановленні програми обираємо всі необхідні модулі для веб-розробки та панель управління (Apache, PHP) в директорію, що не містить кирилических символів. Після встановлення запусить Open Server і оберіть у треї пункт «Запустити всі служби».

У складі Open Server вже налаштовано:

- Веб-сервер Apache з підтримкою .htaccess;
- Систему керування базами даних MySQL 8.4;
- PHP необхідної версії (рекомендовано 8.4);

- phpMyAdmin.

Після встановлення починаємо розгортання веб-додатку. Скопіюйте всі файли проєкту brighthotel в папку домену Open Server: D:\OSPanel\home\. Імпортуйте SQL-дамп бази даних (файл з папки Database/), який містить структуру таблиць rooms, reservation, admin та foods. Перевірте та за потреби відредагуйте параметри підключення до бази даних у файлі includes/db.php:

```
if(!defined('DB_SERVER')) define('DB_SERVER', 'MySQL-8.4');
if(!defined('DB_USERNAME')) define('DB_USERNAME', 'root');
if(!defined('DB_PASSWORD')) define('DB_PASSWORD', '');
if(!defined('DB_NAME')) define('DB_NAME', 'hotel');
```

Після завершення налаштування відкрийте браузер та перейдіть за адресою: <http://brighthotel/>. Для входу в адміністративну панель використовуйте адресу: <http://brighthotel/admin/>.

Висновок до розділу 4

У четвертому розділі було здійснено практичну реалізацію веб-додатку Bright Hotel.

Розроблено клієнтську частину системи з використанням технологій HTML5, CSS3 та JavaScript. Реалізовано всі основні сторінки: головну, номерів, бронювання, «Про нас», блог та контакти. Сайт має сучасний адаптивний дизайн, зручну навігацію та динамічні елементи (слайдери Owl Carousel, графіки Chart.js).

На серверній стороні реалізована архітектура на базі PHP з підключенням до бази даних MySQL [4, 21]. Основний функціонал онлайн-бронювання повністю працездатний: форма приймає дані користувача та зберігає їх у таблицю reservation. Розроблено адміністративну панель, яка дозволяє переглядати бронювання, отримувати статистичні дані та керувати основними процесами [20].

Проведено комплексне тестування системи, яке підтвердило її стабільність, адаптивність та кросбраузерну сумісність [8, 9]. Виявлені недоліки не впливають на загальну працездатність, але вказують на напрямки подальшого вдосконалення.

Також розроблено детальну інструкцію користувача, яка описує процес встановлення та запуску веб-додатку в середовищі Open Server.

Таким чином, у четвертому розділі успішно реалізований функціональний веб-додаток для онлайн-бронювання номерів готелю Bright Hotel, який відповідає поставленим завданням, має сучасний інтерфейс та може бути використаний як для демонстраційних цілей, так і як основа для реального впровадження [19, 26].

Обрана архітектура та технологічний стек повністю виправдали себе при реалізації проєкту [3, 5, 7]. Використання тришарової клієнт-серверної архітектури та процедурного підходу в PHP дозволило швидко розробити стабільну систему, придатну для подальшого розвитку [4, 21]. Розроблений дизайн та фронтенд-реалізація відповідають сучасним вимогам до користувацьких інтерфейсів веб-додатків [20, 25].

ВИСНОВКИ

Метою дипломної роботи була розробка сучасного веб-додатку для онлайн-бронювання номерів готелю Bright Hotel - <https://github.com/qXlyKq/brighthotel>. Поставлена мета була повністю досягнута.

У процесі виконання роботи проведено комплексний аналіз предметної області онлайн-бронювання готельних послуг, вивчено існуючі рішення та їх недоліки. На основі цього сформульовано функціональні та нефункціональні вимоги до системи, спроектовано концептуальну, логічну та фізичну модель бази даних.

Розроблено та реалізовано повнофункціональний веб-додаток з використанням технологій PHP, HTML5, CSS3 та JavaScript. Система включає публічну частину з адаптивним дизайном, зручну форму бронювання без обов'язкової реєстрації, а також адміністративну панель для управління бронюваннями та перегляду аналітики.

Під час тестування підтверджено працездатність основного функціоналу, адаптивність інтерфейсу та кросбраузерну сумісність. Розроблено детальну інструкцію з встановлення та запуску системи в середовищі Open Server.

Практичне значення розробки полягає в тому, що створений веб-додаток може бути безпосередньо впроваджений у діяльність готелю Bright Hotel. Система дозволяє автоматизувати процес бронювання, зменшити ручну роботу адміністраторів, підвищити якість обслуговування клієнтів та знизити залежність від зовнішніх агрегаторів бронювання.

У ході роботи також визначено напрямки подальшого вдосконалення системи: розширення функціоналу адміністративної панелі (редагування бронювань, фільтри, експорт даних), а також інтеграція системи онлайн-оплати.

Таким чином, розроблений веб-додаток Bright Hotel є сучасним, функціональним і конкурентоспроможним програмним продуктом, який успішно вирішує поставлені завдання та демонструє ефективне поєднання теоретичних знань і практичних навичок у сфері веб-розробки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Білоус В. В. Web-програмування на PHP та MySQL : підручник. Київ : Видавничий дім «Слово», 2023. 512 с.
2. Bootstrap 5 Documentation. URL: <https://getbootstrap.com/docs/5.3/> (дата звернення: 15.05.2026).
3. Chart.js Documentation. URL: <https://www.chartjs.org/docs/> (дата звернення: 15.05.2026).
4. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2015. 28 с.
5. ДСТУ ISO/IEC 25010:2015. Системи та програмне забезпечення. Вимоги до якості та її оцінювання (SQuaRE). Модель якості системи та програмного забезпечення. Київ : ДП «УкрНДНЦ», 2015. 42 с.
6. Іванов І. І. Дизайн користувацьких інтерфейсів сучасних веб-сервісів. Київ : Політехніка, 2024. 298 с.
7. Книш В. П. Методологія тестування веб-додатків : навчальний посібник. Київ : КНУ ім. Тараса Шевченка, 2024. 284 с.
8. Ковальчук А. М. Сучасні технології розробки веб-додатків : підручник. Львів : ЛНУ ім. Івана Франка, 2024. 368 с.
9. Крамаренко В. С. Тестування програмного забезпечення : підручник. Київ : КНУ, 2025. 416 с.
10. Красильникова О. М. Автоматизація бізнес-процесів готельних підприємств засобами веб-технологій. Економіка та управління. 2024. № 3. С. 78–92.
11. Літвіненко В. О. Розробка адаптивних веб-додатків для сфери послуг. Науковий вісник НУ «Львівська політехніка». 2024. № 12. С. 134–145.
12. Mozilla Developer Network (MDN Web Docs). URL: <https://developer.mozilla.org/uk/> (дата звернення: 18.05.2026).
13. MySQL 8.4 Reference Manual. URL: <https://dev.mysql.com/doc/refman/8.4/en/> (дата звернення: 20.05.2026).
14. Nielsen J. Usability Engineering. San Diego : Academic Press, 2021. 362 p.

- 15.Омельчук О. П. Автоматизація процесів бронювання в готельному бізнесі. Туризм і готельний бізнес. 2025. № 1. С. 56–72.
- 16.Open Server Panel Official Documentation. URL: <https://ospanel.io/> (дата звернення: 10.05.2026).
- 17.Петренко С. В. Проєктування реляційних баз даних : навчальний посібник. Київ : Політехніка, 2023. 312 с.
- 18.Савченко А. О. Сучасні підходи до розробки веб-систем бронювання. Інформаційні технології та засоби навчання. 2024. № 2. С. 45–61.
- 19.Сучасні тренди розвитку готельного бізнесу в Україні : аналітичний огляд. Київ : Державне агентство розвитку туризму, 2025. 84 с.
- 20.Філіпчук В. Р. Розробка веб-додатку для онлайн-бронювання номерів готелю Bright Hotel : дипломна робота. Херсон : ХНТУ, 2026. 98 с.
- 21.Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley, 2022. 560 p.
- 22.ISO/IEC 12207:2017. Systems and software engineering — Software life cycle processes. Geneva : ISO, 2017. 108 p.
- 23.ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva : ISO, 2011. 34 p.
- 24.PHP: Hypertext Preprocessor. URL: <https://www.php.net/manual/uk/> (дата звернення: 20.05.2026).
- 25.Шаповаленко О. В. Адаптивна верстка веб-інтерфейсів : навчальний посібник. Харків : ХНУРС, 2023. 256 с.
- 26.Чорний С. М. Технології фронтенд-розробки : HTML5, CSS3, JavaScript. Київ : Видавничий дім «Слово», 2024. 428 с.
- 27.Google Web Fundamentals. URL: <https://developers.google.com/web/fundamentals> (дата звернення: 15.05.2026).
- 28.React Documentation. URL: <https://react.dev/> (дата звернення: 18.05.2026).

Оконні форми розроблені в додатку

А.1 - Сторінка бронювання номеру

Забронюйте Номер

Країна, місто...

Оберіть номер ↓

ДАТА ЗАЇЗДУ
ДД.ММ.ГГГГ

ДАТА ВИЇЗДУ
ДД.ММ.ГГГГ

Кількість номерів

Кількість доросл

Кількість дітей ↓

Введіть ім'я

Введіть прізвище

Введіть email

Введіть телефон

ЗАБРОНЮВАТИ

А.2 - Список номерів готеля

BRIGHTHOTEL
Положення
Номери
Бронювання
Про нас
Блог
Контакти



Одномісний номер

\$156 / за ніч

Дорослих: 1 гість

Категорія: Одномісний

Зручності: Шафа, телевізор, телефон, Wi-Fi, ванна кімната

Площа: 20 м²

Тип ліжка: Одне односпальне ліжко



Сімейний номер

\$320 / за ніч

Дорослих: 3 гості

Категорія: Сімейний

Зручності: Шафа, телевізор, телефон, Wi-Fi, ванна кімната

Площа: 30 м²

Тип ліжка: Одне двоспальне та одне односпальне ліжка



Президентський номер

\$425 / за ніч

Дорослих: 1 гість

Категорія: Люкс

Зручності: Шафа, телевізор, телефон, Wi-Fi, ванна кімната, міні-бар

Площа: 45 м²

Тип ліжка: Двоспальне ліжко

Адміністративна панель управління замовленнями

НАЗАД

ID	Країна	Номер	Дата заїзду	Дата виїзду	Кількість номерів	Дорослі	Діти	Ім'я	Прізвище	Електронна пошта	Телефон
1	Україна	Сімейний	2025-11-03	2025-11-09	1	2	1	Костянтин	Бойко	1@ex.com	0987654321
2	Україна	Одинарний	2025-11-04	2025-11-06	1	1	0	Валентина	Корпачук	emma.green@ex.com	09812356745
3	Україна	Президентський	2025-11-26	2025-12-04	1	1	1	Олена	Труханова	123@gmail.com	09988877766
4	Україна	Сімейний	2025-11-24	2025-11-25	1	2	1	VIKTOR	FILIPCHUK	fvr25z@gmail.com	0661944087

ПРО НАС

Наш готель - це поєднання комфорту, сучасного сервісу та затишної атмосфери. Ми створили простір, де кожен гість може віднути турботу, спокій та справжню гостинність. Ми цінуємо ваш час та комфорт, тому пропонуємо продумані умови для відпочинку, роботи та проведення особливих

БЛОГ



Наш розкішний готель і ресторани.

22 травня 2025 4 хвилини



Наш розкішний готель і ресторани.

22 травня 2025 4 хвилини

КОНТАКТНА ІНФОРМАЦІЯ

 Україна, Херсон

 +380123456789

 gmail@tests.com

 Понеділок – П'ятниця, 8:00 - 17:00