

МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА
на тему:

РОЗРОБКА ВЕБ-ДИЗАЙНУ ТА СТВОРЕННЯ ВЕБ-САЙТУ ГОТЕЛЮ

Здобувача вищої освіти першого (бакалаврського) рівня,
4 курсу, групи КБ-19-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»


(підпис)

Меленка Дмитра Олександровича
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Грищенко Світлана Миколаївна,
к.пед.н., ст. досл.
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Керівник освітньо-
професійної програми


(підпис)

Федотова - Півень Ірина
Миколаївна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Допущено до захисту за рішенням кафедри
протокол від «26» травня 2023 № 13

Завідувач кафедри


(підпис)

Філоненко Михайло Миколайович,
к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене
звання)

Ресстрація № 13

05.06.2023

(дата)



(підпис відповідальної особи)

Результати перевірки

до захисту

(до захисту, на доопрацювання)

29.05.2023

(дата)



(підпис)

Результат захисту

5 (відмінно)

(цінка)

15.06.2023

(дата)



(підпис)

Голова комісії:



(підпис)

В.М. Франчук

Члени комісії:



(підпис)

М.М. Філоненко

(підпис)

О.В. Редич

(підпис)

Є.В. Котух

БЛАНК ОЦІНЮВАННЯ ДИПЛОМНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	29		15.06.2023 Франчук В.М. <i>В.М.</i> Філоненко М.М. <i>М.М.</i>
2. Дотримання вимог щодо оформлення	від 0 до 10	9		Редич О.В. <i>О.В.</i>
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		Котух Є.В. <i>Є.В.</i> Суховерхівка Л.П. <i>Л.П.</i>
4. Рецензент	від 0 до 5	5		
РАЗОМ БАЛІВ	від 0 до 50	47		

Оцінювання кваліфікаційної роботи науковим керівником - 45 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підписи)
92	A	5 (відмінно)	15.06.2023	Франчук В.М. <i>В.М.</i> Філоненко М.М. <i>М.М.</i> Редич О.В. <i>О.В.</i> Котух Є.В. <i>Є.В.</i> Суховерхівка Л.П. <i>Л.П.</i>

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

факультет фінансів та цифрових технологій
 кафедра комп'ютерних та інформаційних технологій і систем
 спеціальність 122 «Комп'ютерні науки»
 освітня програма «Інформаційні управляючі системи і технології (за галузями)»

Затверджую

Завідувач кафедри

 Філоненко М.М.
 « 16 » 11 2022

Завдання на кваліфікаційну роботу здобувачу вищої освіти

Мелешку Дмитру Олександровичу

(прізвище, ім'я, по батькові)

Тема випускної кваліфікаційної роботи
розробка веб-дизайну та створення веб-сайту готелю
 затверджена наказом ректора від «15» листопада 2022 № 1205 (зі змінами наказу від «04» квітня 2023 № 361)
 Строк здачі закінченої роботи «26» травня 2023
 Цільова установка застосування набутих знань, умінь в розробці дизайну та веб-сайту готеля
 Мета роботи розробка веб-сайту для покращення процесів бронювання готелів для користувачів та зменшення певних ризиків для бізнесу
 Об'єкт дослідження. Веб-дизайн та веб-розробка
 Предмет дослідження. Процес створення дизайну і реалізація сайту.
 Задачі дослідження веб-сайт готелю Hotel
 Вихідні дані до роботи методи та підходи до розробки та створений веб-сайт
 Перелік графічного матеріалу головна сторінка веб-сайту готелю, сторінка опису готеля, сторінка підтвердження замовлення, базова архітектура веб-сайту, архітектура веб-сайту готеля, схема бази даних веб-сайту готеля
 Зміст випускної кваліфікаційної (дипломної) роботи
 ПІСЬМОВИЙ
 СТРУКТУРА
 РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.
 1. Історія та сучасний стан готельного бізнесу та бізнес-процеси готельної справи в Україні.
 2. Аналіз обраних інструментів розробки веб-сайту готелю.
 РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ВЕБ-САЙТУ.
 1. Дизайн та етапи створення веб-сайту.
 2. Архітектура та база даних веб-сайту готелю.
 РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ГОТЕЛЯ
 ДОДАТКИ

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	За фактом
1	2	3	
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.11.2022	15.11.2022
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	01.02.2023	01.02.2023
3	II розділ (аналіз експериментальної частини про методики досліджень)	03.04.2023	04.04.2023
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	10.05.2023	09.05.2023
5	Висновки	15.05.2022	15.05.2022
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	26.05.2023	26.05.2023
7	Отримання відгуку керівника	06.06.2023	06.06.2023
8	Отримання зовнішньої рецензії	08.06.2023	08.06.2023

8. Дата видачі завдання «16» листопада 2022

9. Керівник випускної кваліфікаційної (дипломної) роботи


 (підпис)
Грищенко С.М.

(прізвище, ініціали)

10. Керівник освітньо-професійної програми


 (підпис)
Федотова-Півень І.

(прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти


 (підпис)
Мелешко Д.О.

(прізвище, ініціали)

АНОТАЦІЯ

Розробка веб-дизайну та створення веб-сайту готелю

Мета роботи полягає в розробці веб-сайту для покращення процесів бронювання готелів для користувачів та зменшення певних ризиків для бізнесу.

Завданнями роботи є:

- Проведення аналізу предметної області.
- Розробку естетичного та привабливого дизайну:
- Створення функціонального веб-сайту
- Впровадження адаптивного дизайну:
- Оптимізація продуктивності та швидкодії:

Предметом дослідження є розробка дизайну та веб-сайту готелю.

Об'єктом дослідження є веб-дизайн та веб-розробка.

Базою дослідження виступає веб-сайт готелю Hotel.

За результатами дослідження сформульовані методика та етапи реалізації веб-сайту.

Одержані результати можуть бути використані при оптимізації бізнес процесів готелю та залученню нових клієнтів завдяки використанню веб-сайту

Кваліфікаційна робота містить 59 сторінок, 6 рисунків, список використаних джерел 40 найменувань, 2 додатка.

Ключові слова: готель, бронювання, номери, онлайн, доступність, зручність, інформація.

Рік виконання кваліфікаційної роботи 2023

Рік захисту роботи 2023

ABSTRACT

Development of web design and creation of a hotel website

The aim of this work is to develop a website to improve the hotel booking process for users and reduce certain risks for the business.

The tasks of the work are as follows:

- Conducting analysis of the subject area.
- Development of aesthetic and attractive design.
- Creation of a functional website.
- Implementation of responsive design.
- Optimization of performance and speed.

The subject of the research is the design and website development of a hotel.

The object of the research is web design and web development.

The research is based on the website of Hotel.

The results of the research include the methodology and stages of website implementation.

The obtained results can be used to optimize hotel business processes and attract new customers through the use of the website.

The qualification work consists of 59 pages, 0 tables, 6 images, a list of 40 references, and 2 appendices.

Keywords: hotel, booking, rooms, online, availability, convenience, information.

Year of qualification work: 2023

Year of work defense: 2023

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Історія та сучасний стан готельного бізнесу та бізнес-процеси готельної справи в Україні	11
1.2. Аналіз обраних інструментів розробки веб-сайту готелю.	19
РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ВЕБ-САЙТУ	27
2.1 Дизайн та етапи створення веб-сайту	27
2.2. Архітектура та проєктування бази даних веб-сайту готелю	38
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ГОТЕЛЯ	47
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТКИ	59

ВСТУП

Сьогодні світ вже важко уявити без використання інтернет-технологій, що свідчить про те, що бізнес розширює свої межі, долучаючи нову аудиторію. Натомість користувач мережі Інтернет отримує більше джерел отримання нової інформації, товарів або послуг.

Кожен день людина користується Інтернетом за допомогою різноманітних гаджетів у пошуку задоволення власних потреб, застосовуючи веб та мобільні додатки. З розвитком швидкості Інтернету зростає кількість охочих залучити інструменти віддаленого доступу та обслуговування. Для підприємств наявність веб-сайту становиться невід'ємною частиною розвитку. Веб-сайт є своєрідним посередником, так званим інтерфейсом, між підприємством та клієнтами.

Стрімкий розвиток веб-технологій набуває поширення в усіх сферах людського життя, зокрема це стосується і сфери надання послуг. Можливість виконати замовлення, не виходячи навіть з власної оселі, привертає увагу сучасного клієнта. Це стосується різних напрямків сфери обслуговування, у тому числі та готельного бізнесу.

Актуальність даної теми полягає в тому, що готельна індустрія є одним із найважливіших компонентів ширшої індустрії послуг, яка обслуговує клієнтів, які потребують ночівлі. Готельна справа здебільшого створює дохід для місцевої економіки безпосередньо, коли туристи відвідують і витрачають гроші в готелях, ресторанах і розважальних закладах. Це опосередковано сприяє зміцненню економіки, оскільки туристи купують роздрібні товари, аптечні товари, сувеніри місцевого виробництва та ремісничі вироби [4].

Вчені розглядали дану проблему в своїх роботах [9, 24-27].

Метою кваліфікаційної роботи є розроблення веб-сайту для покращення процесів бронювання готелів для користувачів та зменшення певних ризиків для бізнесу. Зі сторони користувачів це досягається шляхом надання першим

найпопулярніших пропозицій з найактуальнішою інформацією. Зі сторони бізнесу – можливість оцінювання клієнта за певною рейтинговою системою, підтвердження сплат та надання можливості схвалювати або скасовувати бронювання, якщо майбутній клієнт вже отримував негативні відгуки після попередніх перебувань в заброньованих апартаментах.

Для досягнення мети необхідно вирішити такі *завдання*:

- Проведення аналізу предметної області. Було проаналізовано готельний бізнес та визначено актуальність створення веб-сайту
- Розробку естетичного та привабливого дизайну: В рамках дослідження були вивчені сучасні тенденції у дизайні готельних веб-сайтів та було розроблено оригінальний та привабливий дизайн, який відображає унікальність та атмосферу готелю;
- Створення функціонального веб-сайту: Була проведена розробка та реалізація веб-сайту, що включає основні функціональні елементи, такі як бронювання номерів, перегляд доступних послуг та зручне навігаційне меню;
- Впровадження адаптивного дизайну: Була забезпечена адаптивність веб-сайту, що дозволяє користувачам зручно переглядати та використовувати сайт на різних пристроях, включаючи комп'ютери, планшети та мобільні пристрої.
- Оптимізація продуктивності та швидкодії: Було проведено оптимізацію веб-сайту з метою забезпечення швидкої роботи та мінімізації часу завантаження сторінок.

Об'єктом кваліфікаційної роботи є веб-дизайн та веб-розробка

Предмет дослідження – розробка дизайну та веб-сайту готелю.

Методи дослідження – теоретичні – аналіз, узагальнення, систематизація законодавчої бази, матеріалів конференцій, Інтернет-ресурсів для виділення теоретичних засад дослідження; емпіричні – аналіз досвіду роботи в даній галузі, для констатування стану розв'язання проблеми.

Базою дослідження виступає веб-сайт готелю Hotel.

Наукова-практична значимість одержаних результатів. Сприяє покращенню користувацького досвіду, ефективному бронюванню номерів, підвищенню конкурентоспроможності готелю, оптимізації його роботи та забезпеченню доступної та актуальної інформації для клієнтів. Ці результати дозволяють готелю залучати більше клієнтів, підвищувати ефективність управління та забезпечувати успішну роботу в конкурентному середовищі.

Веб-сайт готелю має велике практичне значення, оскільки веб-сайти стали важливим інструментом для просування послуг готелів та бронювання номерів.

По-перше, розроблений веб-сайт дозволить готелю привернути нових клієнтів і збільшити свій потенційний ринок. Крім того, веб-сайт дозволить готелю підтримувати сталих клієнтів, надаючи їм зручний і легкий спосіб бронювання номерів та отримання інформації про готельні послуги.

По-друге, розробка веб-сайту дозволить готелю підвищити свій рівень конкурентоспроможності на ринку, оскільки сучасні технології веб-розробки дозволяють створювати високоякісні і привабливі веб-сайти, які здатні привернути більше клієнтів.

Апробація результатів роботи. Основні результати дослідження висвітлено в конференції «Цифрова трансформація фінансової системи України та Країн V-4 в умовах євроінтеграції» (м. Дубляни, 26 квітня 2023р.).

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Історія та сучасний стан готельного бізнесу та бізнес-процеси готельної справи в Україні

Готельний бізнес є одним з найбільш перспективних і успішних напрямків бізнесу в Україні. Орієнтація на європейські стандарти обслуговування та активний перехід на такі стандарти є відмінною рисою цього сектору. Як і будь-яка інша економічна діяльність, готельний бізнес прагне збільшити свій дохід і шукає інструменти та ефективні способи досягнення бажаних фінансових результатів.

Перші заклади, подібні за набором послуг до готелів, виникають в Україні в період Київської Русі, тобто в XII – XIII столітті [7]. З'явилися вони завдяки вигідному розташуванню: країни в Середземномор'ї, Прибалтика і Західна Європа оточували Київську Русь. Отже, туристів було багато. У вісімнадцятому столітті почалося будівництво нових та вдосконалених готелів, їх називали «поштовими трактами». Звичайно, їх зовнішній вигляд був далекий від сучасного готелю, але послуги та мета створення таких закладів були дуже схожі. Побудований у 1803-1805 рр. «Зелений готель» був найбільш відомий у 1950-х роках.

До 1880 року у Києві було 15 готелів. В них можна було поїсти, переночувати та залишити свого коня в конюшні, де за ним наглядав персонал готелю. За наступні 20 років побудовано 64 нових готелів в центральній частині міста, переважно в районі Хрещатику та прилеглих до нього вулиць.

До Першої світової війни в Києві було близько 80 готелів. Серед таких готелів найвідомішими були: «Австрія», «Англія», «Брістоль», «Версаль», «Марсель» та інші, назва яких відобразила географію всієї Європи.

До 2014 року готельний бізнес розвивався дуже швидкими темпами. 2014 рік почався з кризи на готельному ринку, через яку багато з них призупинили, а деякі навіть припинили свою діяльність [18].

Протягом останніх років готельний бізнес відновлює свою діяльність і розвивається в нових специфічних напрямках, забезпечуючи більш широкий спектр високоякісних послуг щороку.

Найбільше туристів прибуло в Україну у 2017-2019 роках. У середньому – це 13-15 мільйонів туристів на рік. Потім «прийшла» пандемія і в 2020 році кількість відвідувачів різко скоротилась. За весь рік до країни приїхало 3,3 млн туристів, з яких 1,8 млн у перші 3 місяці року. Однак популярності почав набирати внутрішній туризм. Українці відкрили для себе країну з іншого боку, адже стало з'являтися багато інформації про туристичні центри нашої держави, проводилась активна робота з популяризації курортних міст України та почали мандрувати рідними, але досі незвіданими куточками.

У 2021 році туристичні потоки стали відроджуватись. До України знову поїхали іноземці, країна зустріла 4,3 млн гостей, 70% з яких – це європейці. У цей рік запрацювала програма залучення іноземних мандрівників та відкриття нового туристичного ринку, запроваджена керівництвом України. Завдяки злагодженій роботі влітку 2021 року туристи із Саудівської Аравії створили в Україні справжній туристичний бум, до країни приїхало понад 50 000 гостей з цієї далекої країни. Далі 24 лютого 2022 року близько 5 години ранку Путін оголосив війну Україні. Одразу ж російські війська розпочали інтенсивні обстріли підрозділів ЗСУ на сході країни, перетнули північно-східні кордони, а також нанесли ракетно-бомбові удари по аеродромах і складах зброї майже на всій території України. У зв'язку з цим наразі є суттєва невизначеність щодо подальшого стану туризму України. Деякі експерти вважають, що туризм в Україні зовсім завмер, деякі, що просто змінилась мета та характер подорожей [5].

Незважаючи на війну люди все таки мандрують, заради перезавантаження та зниження рівня тривоги, а деякі українці вперше виїжджають за межі свого населеного пункту та потрапляють туди, куди не могли поїхати багато років через

постійну роботу та нестачу часу. Тому треба вірити, що український туризм живе, але він трохи інший.

В Закарпатській, Львівській, Івано-Франківській та Хмельницькій областях на сьогодні відбувається швидке зростання кількості туристів. За інформацією Державного агентства розвитку туризму (ДАРТ), 1,5 мільйони туристів відвідали західні та північні області України. Як і в Києві, так і в інших містах України більшість гостей готелів складають українці – близько 90% [7].

Сподіваємось, що війна закінчиться якнайшвидше та після останнього пострілу туристична галузь України отримає новий ковток свіжого повітря. Адже на даний час Україна у всіх на вустах, у кожному куточку світу людям дуже кортить подивитись, що це за така мужня та незламна держава, яка у 21 сторіччі, розташовуючись у центрі Європи, опинилась у пеклі, але змогла вистояти завдяки неймовірним та відважним людям. Експерти вважають, що у перші роки після війни Україна буде входити до Топ-5 найвідвідуваніших країн світу, а наша культура, гостинність та краса держави обов'язково зроблять свою справу і туристи будуть повертатись до нас знову й знову.

А отже нам не вистачатиме готелів. Відповідно до різних оцінок, Києву потрібно ще 12-15 тис. готельних номерів. З 8 тисяч готельних номерів столиці тільки 3% відповідають міжнародним стандартам. В той самий час, середня вартість готельних послуг в Києві становить близько 1800 грн на добу, і вона постійно зростає [5].

Представники іноземного капіталу загалом оцінюють український готельний ринок як складний, але перспективний для інвесторів і звертають на нього більше уваги для можливості реалізації своїх проектів у регіонах країни.

Деякі експерти вважають, що після війни швидко активізуються інвестори у готельному сегменті комерційної нерухомості України. Існує кілька причин для цього, зокрема:

- Великий наплив туристів

- Нездатність існуючих готелів задовольнити майбутній попит
- Насичення сегментів ринку нерухомості з мінімумом термін окупності
- Збільшення капіталу інвестиційних компаній
- Зміни у зовнішній політиці
- Наближення кордонів ЄС до кордонів України і поступово інтеграція держави в європейське та світове співтовариство
- Покращення інвестиційного та бізнес-клімату в Україні
- Поступове зростання ділової активності всередині країни
- Збільшення частоти ділових візитів з-за кордону
- Зростання туристичної привабливості України [17].

Зміни до правил обов'язкової сертифікації готельних послуг вже готуються. Нові правила сертифікації можуть вплинути на динаміку готельного бізнесу в майбутньому. Це в свою чергу стане запорукою стабільності та процвітання готельного бізнесу в Україні.

Поняття «бізнес-процеси» останнім часом досить популярно. Бізнес-процес – це набір дій, що виконується в компанії для одержання заданого результату [6]. Відповідно до цього визначення у діяльності готельно-ресторанного комплексу можна виділити наступні групи бізнес-процесів: 1) продажу готельних номерів; 2) обслуговування готелю; 3) організації харчування; 4) постачання готелю; 5) надання додаткових послуг.

Готельно-ресторанний бізнес є основною складовою туристичної галузі нашої держави. Розвиток внутрішнього та міжнародного туризму значною мірою пов'язаний з рівнем матеріально-технічної бази туристичних підприємств, розгалуженістю та різноманітністю їхньої мережі, якістю по обсягу послуг; пропонованих готельним господарством [3]. Готельні підприємства виконують одну з основних функцій у сфері обслуговування туристів – забезпечують їх житлом і побутовими послугами під час подорожі.

Свого часу поняття «готельне господарство» співвідносилося з господарською діяльністю, що полягала в наданні платних послуг розміщення в готелях. Згодом зі зростанням попиту на туристичні послуги та прагненням готелів до розширення комплексності обслуговування, послуги розміщення почали тісно пов'язуватись з харчуванням і реалізацією додаткових послуг. Це дає підстави для визначення поняття «готельне господарство» в широкому та вузькому його розуміннях. Визначення поняття «готельне господарство» у широкому розумінні включає проживання, харчування та додаткові послуги; у вузькому розумінні лише проживання.

Прискорений розвиток туристичної галузі, прагнення підприємств до отримання найвищого прибутку, зростала конкурентоспроможність споживачів послуг сприяють розширенню та урізноманітненню готельної діяльності. Поряд з послугами проживання і харчування пропонуються такі нові додаткові послуги як обслуговування ділових зустрічей, спортивні, медичні, посередницькі, комунально-побутові та ін.. При цьому підприємства готельного господарства розширюють послуги не тільки притаманні даній галузі, а й іншим галузям, наприклад, харчовій.

Економічна сутність готельної діяльності полягає у тому, що вона має нематеріальний характер. Результатом виробничо-експлуатаційної діяльності готелів є не «готовий продукт», а пропозиція особливого виду послуг. Разом з тим, послуги не можуть надаватись окремо від наявного матеріального продукту, тобто без експлуатації матеріально-технічної бази (будівлі, споруди, устаткування, інвентар), що виступає основою виробництва та реалізації пропонованих послуг. З урахуванням особливостей обслуговування в готельному господарстві, де поєднуються виробництво та споживання послуг, цей процес визначається поняттям «надання послуг». Слід підкреслити, що в процесі цієї діяльності здійснюється продаж і безпосереднє надання послуг.

Рівень обслуговування в сучасному готельному господарстві зумовлений в основному станом матеріально-технічної бази підприємства, що його пропонує (будівлі, споруди, інженерне й санітарно-технічне обладнання, електроприлади, меблі тощо). В останні роки спостерігається процес оновлення та урізноманітнення матеріально-технічної бази підприємств, підвищення фахового рівня персоналу готелів, впровадження інноваційних технологій тощо. Будівлі готелів обладнуються спортивними, медичними, розважальними та іншими приміщеннями. Відтак процес обслуговування споживачів послуг стає різноманітнішим, а пропоновані послуги набувають комплексного характеру [9].

Поряд з цим, попит на готельні послуги є нерівномірним, що зумовлюється економічними, демографічними, природо-кліматичними факторами. Дуже часто він залежить від сезону, дня, місяця і навіть тижня. Тому готелі, мотелі, кеші повинні мати певний резерв номерного фонду, а також матеріальних і трудових ресурсів, щоб оперативно і ефективно реагувати на його коливання. Постійна готовність підприємств готельного господарства до зустрічі й обслуговування гостей передбачає необхідність одночасної присутності протягом доби адміністратора, регістратора, портьє, носильників, покоївок та інших фахівців. Це значною мірою знижує ефективність використання праці персоналу. Однак вітчизняний та закордонний досвід відкриває шляхи до подолання цього протиріччя (наприклад, запровадження бригадної організації праці за принципом суміщення професій). Так у готельному об'єднанні «Формула-1», якому належать 200 мотелів на території Франції, Бельгії, Німеччини, Великобританії та Нідерландів у результаті впровадження електронної системи резервування місць, чисельність постійно робочого персоналу в окремих підприємствах скоротилась до мінімуму й становить 2-3 особи на одне підприємство.

Процес надання основної послуги у готельній діяльності можна схематично поділити на кілька основних етапів: інформація про надання комплексу послуг,

резервування, (бронювання місць), реалізація послуг (зустріч і проводи гостей), обслуговування під час перебування в готелі.

Вказані етапи надання послуг проживання є взаємопов'язаними та взаємообумовленими, кожен окремо й усі разом вони визначені як рівень якості діяльності підприємства [16].

З метою підвищення рівня якості та культури обслуговування, а також конкурентоздатності на світовому ринку підприємства готельно-ресторанного бізнесу повинні пропонувати не лише високий рівень комфорту, але й ширший спектр додаткових послуг. Сервіс готелю доцільно формувати не за принципом попиту, а за принципом пропозиції.

Асортимент додаткових послуг у цій галузі постійно розширюється. Якщо раніше в переліку готельних підприємств нашої країни їх значилось не більше десяти, то нині їх вже понад шістдесят, що значно ускладнює роботу менеджерів цієї сфери діяльності.

Додаткові послуги, що їх надають вітчизняні готельні підприємства, за загальними ознаками можна поділити на дві групи, а саме:

Послуги для забезпечення комфортності перебування гостей у готелі та в районі їхнього тимчасового перебування:

□ інформаційні (інформування про послуги, пропоновані готелем, про розташування готелю, транспортне обслуговування, пропозиції щодо придбання товарів, сувенірів тощо», наявність історико-культурних пам'яток і культурно-просвітницьких установ);

□ комунально-побутові (прання, чищення, прасування одягу, ремонт взуття, годинників, валіз, спортивного інвентарю тощо);

□ посередницькі (резервування місць у культурно-розважальних установах, замовлення театральних квитків, резервування номерів в інших місцях перебування тощо);

□ автотранспортні послуги (гараж, стоянка, дрібний ремонт 96 транспортних засобів);

□ продаж товарів, видань ЗМІ, сувенірів тощо; прокат спортивного, пляжного інвентарю, комп'ютерного обладнання, автомобілів, розмножувальної техніки тощо; зберігання кореспонденції, цінних речей [26].

Послуги щодо поінформування мандрівників про район, країну перебування (організація екскурсій, зустрічей, перегляд фільмів, участь в урочистостях з нагоди національних свят тощо);

Послуги, що передбачають підвищення рівня комфорту в готельних номерах (харчування в номері, встановлення додаткового обладнання тощо); Послуги, пов'язані із задоволенням особливих побажань гостей, діячів мистецтва, спортсменів, бізнесменів та ін. (користування музичними інструментами, спортивним інвентарем, послуги особистих екскурсоводів перекладачів, секретарів, забезпечення телексом зв'язком» комп'ютерним обладнанням тощо).

Залежно від виду та категорії готелю перелік додаткових послуг значно відрізняється. Готелі категорії «5 зірок» повинні надавати всі додаткові послуги самотійно через широку мережу власних підрозділів (перукарня, ресторан, кафе, бар, відхилення зв'язку та банку, магазини, кіоски, бюро різних видів послуг тощо) [11].

Підприємства готельно-ресторанного бізнесу високої категорії, як правило, мають високу місткість, більш низької - малу або середню за обсягом. Саме тому вони мають різну чисельність управлінського персоналу. Утримання великого апарату управлінців економічно виправдане лише на великих підприємствах. У зв'язку з ліквідацією централізованих управлінських структур, набуттям повної господарської самостійності, розгортанням приватизаційних процесів керівники підприємств змушені самотійно планувати їхній подальший розвиток, укласти відповідні господарські угоди, забезпечувати технологічні процеси надання

послуг, підбирати кадри і керувати роботою персоналу, щоденно вирішувати безліч непередбачуваних проблем.

Значно збільшився обсяг управлінської праці, пов'язаної з постачанням, обліком, контролем виконання завдань. Все це потребує високої кваліфікації керівників, їхнього вміння на високому рівні виконувати будь-які управлінські функції [37].

Розглянуті особливості менеджерської діяльності підприємств готельно-ресторанного спрямування спонукають їх до об'єднання в асоціації, господарські товариства, де ряд управлінських функцій можна виконувати централізовано (укладення угод щодо постачання, ремонту, маркетингових досліджень, правового консультування, підготовки кадрів з обліку роботи тощо).

Готельний бізнес - це нелегко, але при правильному менеджменті і хорошему маркетингу, готель здатний швидко окупитися і почати приносити прибуток. Готельний ринок України має великі перспективи в майбутньому та буде одним із найперспективніших у Східній Європі, але з усім потенціалом для розвитку готельного бізнесу.

1.2. Аналіз обраних інструментів розробки веб-сайту готелю.

В даний час у сфері розробки програмного забезпечення постає безліч завдань, починаючи від створення інтерактивних розважальних сайтів до серйозних бізнес-проектів, які вимагають підвищеної надійності та безпеки від несанкціонованого доступу. Для успішної реалізації цих завдань важливо правильно вибрати необхідні інструменти, такі як мови програмування, фреймворки та системи управління контентом, які стають все більш актуальними.

На сьогоднішній день доступно багато мов програмування, які можуть бути використані для розробки веб-додатків. Для розробки веб-сайту готелю було обрано такі інструменти, як мова гіпертекстової розмітки HTML, мова для запису

стильового оформлення веб-сторінок CSS, модуль SCSS, мова програмування JavaScript та фреймворк React [33].

HTML є скороченням від HyperText Markup Language. Це стандартна мова розмітки для створення веб-сторінок, яка дозволяє створювати та структурувати різноманітні елементи, такі як розділи, абзаци та посилання, за допомогою HTML-елементів (будівельних блоків веб-сторінки), таких як теги та атрибути.

HTML має багато варіантів використання, а саме:

- Веб-розробка. Розробники використовують HTML-код для проєктування того, як браузер відображає елементи веб-сторінки, такі як текст, гіперпосилання та медіа-файли.
- Інтернет-навігація. Користувачі можуть легко переходити та вставляти посилання між пов'язаними сторінками та веб-сайтами, оскільки HTML широко використовується для вставлення гіперпосилань.
- Веб-документація. HTML дає змогу впорядковувати та форматовувати документи, подібно до Microsoft Word [34].

CSS означає мову каскадних таблиць стилів і використовується для стилізації елементів, написаних на мові розмітки, такої як HTML. Вона відокремлює вміст від його візуального представлення на веб-сторінці. CSS тісно пов'язаний з HTML, оскільки ця мова є основою сайту, тоді як CSS забезпечує його естетичний вигляд.

Хоча використання CSS технічно не є обов'язковим, веб-сторінки без нього матимуть лише елементи HTML та виглядатимуть непривабливо, що може знизити кількість відвідувачів сайту.

Недостатня завантаженість веб-сторінки та неправильний колір фону можуть свідчити про те, що CSS завантажилася неправильно або взагалі не була використана. Без використання CSS всі стилізації повинні бути включені

безпосередньо в HTML-код, що означає, що розробники повинні окремо описувати розмір, колір та інші властивості кожного елемента [33].

CSS дозволяє стилізувати все в іншому файлі, таким чином створюючи дизайн і пізніше інтегруючи файли CSS поверх розмітки HTML. Це робить фактичну розмітку HTML набагато чистішою та простішою для обслуговування.

Узагальнено, можна сказати, що застосування функцій CSS зменшує необхідність повторювати опис вигляду окремих елементів. Це дозволяє скоротити код та зменшити кількість можливих помилок.

CSS також дозволяє використовувати кілька стилів на одній HTML-сторінці, що забезпечує безмежні можливості налаштування. На сьогоднішній день це стає необхідністю, а не лише корисним інструментом.

SCSS – це вдосконалена версія CSS. SCSS був розроблений Hampton Catlin і розроблений Chris Eppstein і Natalie Weizenbaum. Завдяки розширеним функціям його часто називають Sassy CSS. Файл SCSS має розширення .scss [30].

SCSS містить усі функції що і в CSS та, яких немає в CSS, що робить його хорошим вибором для розробників. Також SCSS має багато розширених функцій, SCSS пропонує змінні, ви можете скоротити свій код за допомогою змінних, це велика перевага перед звичайним CSS.

JavaScript – це текстова мова програмування, яка використовується як на стороні клієнта, так і на стороні сервера, що дозволяє зробити веб-сторінки інтерактивними. Якщо HTML і CSS є мовами, які надають структуру та стиль веб-сторінкам, JavaScript надає веб-сторінкам інтерактивні елементи, які залучають користувача. Включення JavaScript покращує взаємодію з веб-сторінкою, перетворюючи її зі статичної сторінки на інтерактивну.

JavaScript в основному використовується для веб-додатків і веб-браузерів. Але JavaScript також використовується за межами Інтернету в програмному забезпеченні, серверах і вбудованих елементах керування обладнанням.

Однією з основних функцій JavaScript є додавання динамічності веб-сторінкам. Це включає відображення анімації, зміну видимості тексту та створення викидних меню.

JavaScript є невід'ємною частиною сучасного веб-розроблення і дозволяє створювати інтерактивні веб-сторінки. За допомогою JavaScript користувач може взаємодіяти з веб-сторінками, що покращує їх функціональність та користувацький досвід. Крім того, JavaScript дозволяє змінювати вміст HTML та атрибути елементів без перезавантаження веб-сторінки. Таким чином, JavaScript допомагає збільшити швидкість та ефективність розробки веб-сайтів та забезпечує їх більш гнучке та динамічне відображення.

JavaScript має низку переваг, які роблять його кращим вибором, ніж його конкуренти, а саме:

- Простота структури, яка спрощує вивчення та впровадження JavaScript, а також він працює швидше, ніж деякі інші мови.
- Швидкість (JavaScript виконує сценарії безпосередньо у веб-браузері без попереднього підключення до сервера чи компілятора. Крім того, більшість основних браузерів дозволяють JavaScript компілювати код під час виконання програми).
- Універсальність (JavaScript сумісний з іншими мовами, такими як PHP, Perl і Java).
- Популярність (доступно багато ресурсів і форумів, для оволодіння технічними навичками та знаннями JavaScript).
- Навантаження на сервер (JavaScript зменшує кількість запитів, які надсилаються на сервер. Перевірку даних можна виконати через веб-браузер, а оновлення стосуються лише певних розділів веб-сторінки).

- Оновлення (команда розробників JavaScript і ECMA International постійно оновлюють і створюють нові фреймворки та бібліотеки, забезпечуючи їх актуальність у галузі) [35].

Як і будь-яка інша мова програмування, JavaScript має кілька обмежень, які потрібно враховувати, а саме:

- Сумісність браузера (різні веб-браузери по-різному інтерпретують код JavaScript, що спричиняє неузгодженість).
- Безпека (код JavaScript, який виконується на стороні клієнта, вразливий до використання безвідповідальними користувачами).

Сьогодні інтерфейсні фреймворки та бібліотеки стають невід'ємною частиною сучасної веб-розробки. React.js – це інтерфейсна бібліотека, яка поступово стала основним фреймворком для сучасної веб-розробки в спільноті JavaScript [35].

«Фреймворк React.js – це фреймворк і бібліотека JavaScript із відкритим кодом, розроблена Facebook. Він використовується для швидкого й ефективного створення інтерактивних інтерфейсів користувача та веб-додатків із значно меншою кількістю коду, ніж JavaScript» [25].

У React, при розробці програм, використовуються повторно використовувані компоненти, які можна порівняти з незалежними блоками Lego. Ці компоненти є окремими частинами кінцевого інтерфейсу, які разом утворюють весь інтерфейс користувача програми.

Основна функція React полягає у відображенні рівня представлення додатку, подібно до V у шаблоні "model-view-controller" (MVC), щоб забезпечити ефективну візуалізацію. React заохочує розробників розділяти складні інтерфейси користувача на окремі компоненти, які можуть бути повторно використані, замість роботи з усім інтерфейсом користувача як єдиним блоком. Цей підхід дає змогу швидко та ефективно створювати високодинамічні та адаптивні веб-додатки. Крім того, фреймворк ReactJS поєднує швидкість та ефективність

JavaScript з ефективним методом маніпулювання DOM для швидкого відтворення веб-сторінок..

Зазвичай, для отримання доступу до веб-сторінки ви вводите URL-адресу веб-сторінки в веб-переглядач. Після цього, ваш веб-переглядач надсилає запит на сервер з метою отримати цю веб-сторінку для подальшого відображення. Якщо ви переходите на іншу сторінку веб-сайту за допомогою гіперпосилання на поточній сторінці, то ваш веб-переглядач знову надсилає запит на сервер для отримання цієї нової сторінки.

Даний процес повторюється для кожної нової сторінки чи ресурсу, до якого ви звертаєтесь на веб-сайті. Хоча такий підхід до завантаження веб-сайтів зазвичай є ефективним, він не є оптимальним для веб-сайтів, які мають велику кількість динамічної інформації. Наприклад, попереднє завантаження повної веб-сторінки для веб-сайту з великою кількістю даних було б неефективним та призвело б до нестабільної взаємодії з користувачем.

Крім того, коли дані змінюються в традиційній програмі JavaScript, для відображення цих змін потрібно ручне маніпулювання DOM. Ви повинні визначити, які дані змінилися, і оновити DOM, щоб відобразити ці зміни, що призведе до повного перезавантаження сторінки.

React використовує інший підхід, дозволяючи створювати так звану односторінкову програму (SPA). Односторінкова програма завантажує лише один документ HTML за першим запитом. Потім він оновлює певну частину, вміст або тіло веб-сторінки, які потребують оновлення, за допомогою JavaScript [27].

Цей шаблон відомий як маршрутизація на стороні клієнта, оскільки клієнту не потрібно перезавантажувати повну веб-сторінку, щоб отримати нову сторінку кожного разу, коли користувач робить новий запит. Натомість React перехоплює запит і лише отримує та змінює ті розділи, які потрібно змінити, без необхідності ініціювати повне перезавантаження сторінки. Цей підхід забезпечує кращу продуктивність і більш динамічний досвід користувача.

React покладається на віртуальний DOM, який є копією фактичного DOM. Віртуальний DOM React негайно перезавантажується, щоб відобразити цю нову зміну щоразу, коли відбувається зміна стану даних. Після цього React порівнює віртуальний DOM із фактичним DOM, щоб зрозуміти, що саме змінилося.

Після цього React знаходить найменш дорогий спосіб виправити фактичний DOM за допомогою цього оновлення без відтворення фактичного DOM. Як наслідок, компоненти та інтерфейс React дуже швидко відображають зміни, оскільки вам не потрібно перезавантажувати всю сторінку кожного разу, коли щось оновлюється [25].

«WebStorm – це інструмент розробки, спеціально розроблений для веб-розробки. Він містить усі інструменти, необхідні для створення, редагування та налагодження коду в JavaScript, HTML, CSS, XML та інших пов'язаних з Інтернетом мовах. WebStorm також підтримує кілька різних фреймворків і бібліотек, включаючи Angular, React, Vue.js і Node.js. Інструмент також містить інтелектуальний менеджер вихідного коду, автоматичне завершення коду, візуальний налагоджувач і підтримку кількох версій JavaScript» [35].

Плюси WebStorm, IDE JetBrains для JavaScript

1. Високий рівень продуктивності: WebStorm пропонує широкий спектр функцій, що дозволяє розробникам швидко і ефективно створювати вебдодатки. Наприклад, веб-розробники можуть використовувати автодоповнення коду для зменшення кількості набираєних символів та підвищення продуктивності розробки.
2. Підтримка різних фреймворків: WebStorm підтримує різні фреймворки веб-розробки, включаючи Angular, React, Vue.js та Node.js. Це дозволяє розробникам працювати з будь-якими фреймворками в зручному інтерфейсі розробки.
3. Аналіз коду та налагодження: WebStorm пропонує потужні інструменти для аналізу коду та налагодження. Наприклад, ви можете використовувати різні

типи аналізу коду, щоб виявити помилки в ранній стадії розробки та запобігти проблемам в подальшому.

4. Підтримка інструментів збирання та пакування: WebStorm підтримує різні інструменти збирання та пакування, такі як webpack, Gulp та Grunt. Це дозволяє розробникам легко налаштувати процес збирання та пакування веб-додатків в зручному інтерфейсі.

5. Підтримка Git: WebStorm має вбудовану підтримку системи контролю версій Git, що дозволяє розробникам легко працювати з Git в інтерфейсі розробки.

6. Підтримка інших технологій та інструментів: WebStorm підтримує різноманітні технології та інструменти, що робить його потужним інструментом для розробки веб-додатків. Він працює з різними фреймворками, такими як React, Angular, Vue.js, і надає розширену підтримку TypeScript, CoffeeScript, Less, Sass, і багатьох інших технологій.

Незважаючи на те, що WebStorm має багато переваг, є кілька потенційних мінусів:

1. Вартість: WebStorm - це комерційний продукт, що означає, що для його використання необхідно придбати ліцензію. Це може бути додатковим навантаженням на бюджет розробника, особливо для початківців.

2. Вимогливість до ресурсів: WebStorm - це великий і потужний продукт, який може бути вимогливим до ресурсів комп'ютера. На старих або слабших комп'ютерах це може призвести до зниження продуктивності.

3. Вивчення: WebStorm має багато функцій і може здаватися складним для початківців. Розробникам, які тільки починають використовувати IDE, може знадобитися деякий час, щоб вивчитися всім функціям та інструментам.

4. Підтримка: JetBrains надає гарну підтримку для своїх продуктів, проте деякі проблеми можуть залишатися невирішеними довгий час.

5. Незручності при роботі з іншими мовами програмування: WebStorm призначений для роботи з JavaScript та іншими мовами, що використовуються в

веб-розробці. Якщо вам потрібно працювати з іншими мовами програмування, то WebStorm може бути не найкращим варіантом.

Отримані результати свідчать про те, що використання технологій React, Javascript, WebStorm, HTML та CSS дозволяє створювати високоякісні веб-сайти з відповідним рівнем функціональності та дизайну. Використання WebStorm як інтегрованого середовища розробки (IDE) дозволяє зручно та ефективно розробляти та тестувати вебдодатки.

Крім того, React є досить популярним та зручним інструментом для розробки веб-додатків, оскільки він дозволяє створювати складні інтерфейси користувача за допомогою компонентів та забезпечує високу продуктивність.

Таким чином, обрані інструменти дозволяють ефективно розробляти веб-сайт готелю з відповідним рівнем функціональності та дизайну, що відповідає сучасним вимогам інтернет-простору.

РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ВЕБ-САЙТУ

2.1 Дизайн та етапи створення веб-сайту

Процес розробки веб-сайту є системним набором складних та передбачуваних кроків, які повинні бути виконані для успішного завершення проєкту. Цей процес забезпечує узгодженість ресурсів розробки, зацікавлених сторін та членів команди, щоб гарантувати, що всі аспекти проєкту розглядаються та виконуються вчасно.

Створення веб-сайту складається з двох основних етапів: розробки та дизайну. Процес також може включати програмування веб-сайту, публікацію веб-сайту та керування базами даних.

Веб-розробка стосується функціональності веб-сайту, тоді як веб-дизайн описує зовнішній вигляд веб-сайту. Обидва аспекти є однаково важливими навичками, коли справа доходить до створення веб-сайту. Найкращий веб-сайт, ідеально підібраний для вашого бренду, не буде успішним, якщо він не функціонує належним чином. Зворотно, веб-сайт, який працює добре, але має незадовільний дизайн, також не буде успішним [32].

Процес розробки веб-сайтів має індивідуальний характер і складається з восьми основних кроків. Починаючи з першого кроку, необхідно здійснити збір інформації, що передбачає визначення мети, основних цілей та цільової аудиторії веб-сайту.

Мета веб-сайту визначає його призначення та завдання. Цілі веб-сайту вказують на бажані результати, що повинні бути досягнуті завдяки його створенню. Наявність цілей допомагає визначити зміст та функціонал веб-сайту. Цільова аудиторія визначається як група користувачів, яка буде звертатися до даного веб-сайту. З'ясування цільової аудиторії є ключовим етапом, що допомагає визначити побажання та очікування користувачів.

Другий крок – планування, передбачає створення карти сайту на основі зібраної інформації. Карта сайту включає в себе перелік тем та підтем веб-сайту і допомагає візуалізувати користувацький досвід та можливості переходу між різними сторінками веб-сайту. Успішне планування веб-сайту є необхідним етапом для створення привабливого та функціонального веб-сайту з легкою навігацією.

Третій етап проектування веб-сайту передбачає розробку дизайну, який включає в себе весь візуальний контент, такий як фотографії та відео. Під час розробки дизайну необхідно враховувати цільову аудиторію, яку слід відобразити в інтерфейсі користувача для досягнення оптимального досвіду користувача [31].

Макет веб-сайту є результатом роботи дизайнера і може бути як графічним ескізом, так і повноцінним графічним дизайном. Його основною функцією є візуалізація інформаційної структури та демонстрація основної функціональності. У процесі створення макету використовуються такі елементи, як кольори, логотипи та зображення, що дозволяє отримати загальне уявлення про майбутній продукт.

Часто компанії звертаються до дизайнерів UI/UX з аутсорсингових компаній для розробки макету сайту та створення оптимального інтерфейсу користувача. Після виконання роботи клієнт має можливість переглянути макет і надіслати свій відгук. У разі потреби дизайнер вносить зміни в макет, поки клієнт не буде повністю задоволений. Такий цикл роботи дозволяє отримати оптимальний результат та задовольнити вимоги замовника.

Четвертий етап – контент, розміщений на веб-сайті, є одним з найважливіших компонентів веб-сайту. Ваш контент буде передавати повідомлення аудиторії та стимулювати їх до використання вашого веб-сайту. Написання контенту також передбачає створення привабливих заголовків, редагування тексту, написання нових блоків контенту та інших елементів, що потребують часу та зусиль. Зазвичай клієнт надає готовий до міграції на веб-сайт

контент. Найкраще, якщо весь контент надається до або під час розробки веб-сайту.

До п'ятого етапу відноситься інформаційна архітектура. Вона є структурою та ієрархією організації інформації на вашому веб-сайті. Корисно візуалізувати цю структуру, щоб мати уявлення про розташування контенту та інформації на веб-сайті або вебдодатку.

Щоб визначити необхідний обсяг робіт, потрібно спочатку зрозуміти структуру веб-сайту. Часто структура веб-сайту розвивається з часом та зазвичай призводить до неузгодженості взаємодії з користувачем та зниження конверсій. Оцінюючи та, потенційно, реструктуруючи інформаційну архітектуру, ми забезпечуємо веб-сайт, структурований для користувачів та пошукових систем, та розробляємо дорожню карту для побудови та структурування веб-сайту у майбутньому [8].

На шостому етапі розпочинається процес створення веб-сайту, що передбачає об'єднання всіх попередніх кроків з метою розробки зовнішнього вигляду та функціональності веб-сайту. Одним із головних завдань є створення зручного та легкого у використанні веб-сайту для користувачів. Зазвичай, розробка відбувається з початку створення домашньої сторінки, а потім продовжується на підсторінки. Також слід забезпечити можливість доступу до веб-сайту як з комп'ютера, так і з мобільного телефону.

Сьомий етап – тестування. Після успішного створення веб-сайту він не може бути одразу запущений. Необхідним є процес тестування, який є важливою складовою етапу введення веб-сайту в експлуатацію. Головною метою тестування є переконання у коректній роботі веб-сайту. Під час цього етапу необхідно перевірити всі посилання та кнопки на веб-сайті, виконати перевірку правопису та переконатися, що веб-сайт відображається на різних пристроях уніфіковано, зокрема на комп'ютері та мобільному телефоні.

Восьмим етапом є запуск веб-сайту, що передусе його ретельному перевірненню та тестуванню кілька разів. Для запуску веб-сайту необхідно завантажити його на сервер, після чого рекомендується виконати останній тест, щоб переконатися у правильній роботі веб-сайту. Після цього веб-сайт буде доступний для загального перегляду.

Дев'ятим етапом є моніторинг та оновлення веб-сайту після його успішного запуску. Помилки можуть траплятися навіть після запуску веб-сайту, тому важливо періодично повертатися до нього та перевіряти його стан. Слід стежити за веб-сайтом, щоб переконатися, що він залишається в хорошому стані. У випадку виникнення будь-яких проблем слід вирішувати їх якомога швидше та завжди підтримувати веб-сайт в актуальному стані.

У процесі створення веб-сайту готелю необхідно пройти кілька етапів розробки, починаючи зі збору інформації та планування, переходячи до дизайну та розробки, тестування та запуску, а також моніторингу та оновлення. Дотримання ідеального процесу розробки гарантує виконання проєкту від початку до кінця та забезпечує його просування в короткостроковій та довгостроковій перспективі [24]. Важливо стежити за веб-сайтом, щоб переконатися, що він залишається в хорошому стані, вирішувати будь-які проблеми та завжди підтримувати свій веб-сайт в актуальному стані. Кожен етап розробки веб-сайту є важливим та необхідним для досягнення успішного результату, який забезпечить ефективне просування готелю в Інтернеті та залучення більшої кількості клієнтів.

Хоча можуть відбуватися відмінності в процесі, принципи залишаються сталими. Дотримання перевіреного та дієвого процесу розробки веб-сайту дозволяє отримати уявлення про необхідні етапи як у короткостроковій, так і у довгостроковій перспективі, що гарантує успішне виконання проєкту.

Підтримання знань та розуміння ідеального процесу розробки веб-сайту забезпечує ефективне виконання проєкту розробки веб-сайту від початку до кінця.

У наші дні веб-сайт є необхідністю для компаній, які бажають залишатися конкурентоспроможними на ринку. Веб-сайт є цифровою візитною карткою, яка дозволяє брендам та компаніям залучати потенційних клієнтів, надавати цінну інформацію про свої продукти та послуги, а також збільшувати коефіцієнти конверсії. Однак, не всі веб-сайти рівні у своєму функціонуванні та ефективності. Щоб веб-сайт успішно виконував свої функції, він повинен бути добре розробленим та зрозумілим для користувачів. Особливу увагу потрібно приділити дизайну, який повинен залучати та утримувати увагу відвідувачів на онлайн-платформі.

Легко помітна панель меню, розташована в верхній частині домашньої сторінки веб-сайту, є ілюстрацією логічного веб-дизайну в дії. Проте, щоб створити видатний веб-сайт, необхідно врахувати набагато більше факторів, ніж лише панель меню. Однак, що саме робить дизайн веб-сайту відмінним? Науково доведено, що деякі хитрощі покращують зручність користування сторінкою. Цей підручник охоплює такі питання:

У сучасну епоху, коли більшість бізнесу ведеться онлайн, добре розроблений веб-сайт стає необхідністю, а не просто "приємністю". Навіть незначна деталь дизайну, така як чітка панель меню, може вирішувати встановлення або порушення відносин з потенційним клієнтом.

Без цього маленького, але важливого доповнення, веб-відвідувачеві може бути складно знайти потрібну веб-сторінку. У найгіршому випадку відвідувач може настільки розчаруватися, що покине веб-сайт - це стане втратою можливості для бренду або бізнесу, які сподіваються залучити цього потенційного клієнта [28].

Важливість веб-дизайну відображається у низці статистичних даних. Наприклад, користувачі приблизно за 50 мілісекунд здійснюють вибір, залишатися на веб-сайті чи переходити далі. Таким чином, 38% людей покидають веб-сайт, якщо вони вважають, що макет або вміст не естетично привабливі.

Варто зауважити, що веб-дизайн і веб-розробка – це дві різні області. Якщо дизайн стосується зовнішнього вигляду веб-сайту (його форма), то розробка займається його функціональністю. Веб-розробка відповідає за процеси, що відбуваються за лаштунками (наприклад, кодування, яке визначає, як веб-сайт реагує на дії користувача), тоді як веб-дизайн – це те, що ви бачите зі спереду та є його центральною складовою.

У кінцевому підсумку найкращі практики доступного веб-дизайну складаються з кількох базових елементів дизайну, які всі спрямовані на покращення взаємодії з користувачем, який переглядає веб-сайт. Серед них особливо важливим є зручна навігація веб-сайту. Ця практика включає основну панель навігації (або меню), яка містить чітко сформульовані пункти, що повідомляють користувачеві, які розділи можна знайти, натиснувши на відповідну сторінку (наприклад, «послуги», «контакти», «про нас»). Ці покажчики мають бути зрозумілими, простими та відповідати контексту вашого веб-сайту. Крім того, їх зручно візуалізувати, забезпечивши чіткий шрифт на простому фоні. Корисно також додатково включити веб-сайт навігаційну карту в нижній колонтитул, строк пошуку для швидкого пошуку ключових слів та навігаційні крихти на кожній сторінці, які дозволяють користувачам легко визначити місцезнаходження на веб-сайті та повторювати кроки, які вони здійснили в разі потреби.

Також важлива мінімізація тексту. На екранах цифрових пристроїв, особливо на планшетах та мобільних пристроях, великі фрагменти тексту мають тенденцію відвертати увагу користувача. Користувачі, які відвідують веб-сайт, зазвичай мають вже на увазі свою місію, таку як отримання інформації або пошук продукту, і не бажають витратити час на читання великої кількості слів. Їм потрібна лише мінімальна інформація, яка допоможе досягти їхньої мети.

Послідовна колірна схема є невід'ємною частиною будь-якої сильної візуальної ідентичності бренду. Такі відомі бренди, як Coca-Cola, Walmart та

McDonald's, мають свої фірмові кольори, які безпосередньо асоціюються з їхньою маркою. У веб-дизайні важливо враховувати колірну схему бренду [30].

Професійний веб-дизайнер знатиме, які кольори гармоніюють між собою і може запропонувати кольорові палітри, які можуть бути використані на всьому веб-сайті та в інших матеріалах для брендування. Навіть протилежні кольорові відтінки можуть гармонійно поєднуватись, що може бути дивовижним для людей без досвіду у дизайні.

Насамперед слід відзначити, що відсутність кольору, або так званий «білий простір», є надзвичайно важливим елементом процесу веб-дизайну, нарівні з узгодженою колірною схемою. Негативний простір, який знаходиться між елементами сторінки, полями, жолобами та проміжною графікою, також є важливою складовою частиною дизайну. Його використання допомагає підтримувати веб-сайт у візуально привабливому та збалансованому стані [28].

Окрім того, візуальні елементи та зображення є невід'ємною складовою будь-якого веб-сайту. Вони допомагають збалансувати текст, зробити сторінку більш привабливою та переконливою. Візуальні матеріали можуть бути набагато ефективнішими, оскільки люди часто краще засвоюють інформацію через зорове сприйняття, аніж через читання. Крім того, деякі люди краще сприймають візуальні зображення, ніж текстовий контент.

Щоб забезпечити ефективне використання візуальних елементів на веб-сайті, можна скористатися багатьма різними типами візуальних елементів, таких як фотографії, інфографіка, відео, зображення екрана та анімація. Важливо не перевантажувати вебсторінку візуальним контентом, але забезпечити достатню різноманітність, щоб зберегти її цікавою та привабливою для відвідувачів.

Динамічний контент, також відомий як адаптивний вміст, є формою вмісту, який автоматично змінюється відповідно до поведінки користувача. Це забезпечує більш персоналізований та зручний досвід перегляду, що в загальному плані забезпечує більше задоволення. Один з прикладів динамічного вмісту – цільова

сторінка, яка адаптується до місцезнаходження користувача на основі IP-адреси. Зовнішній вигляд цільової сторінки України може, наприклад, відрізнятися від сторінки США або ЄС.

Динамічний вміст може значно сприяти підвищенню конверсій. Наприклад, сайт електронної комерції може запропонувати користувачеві персоналізовані рекомендації на основі попередніх переглядів або покупок. Однак, реалізація динамічного вмісту вимагає уваги не тільки до веброзробки, але й до дизайну, об'єднуючи «задні» та «передні» операції для створення загального адаптивного дизайну.

Дизайн веб-сайту є важливою складовою процесу розробки веб-сайту готелю. Головною метою дизайну веб-сайту є забезпечення зручного та простого користування сайтом клієнтами готелю. Для досягнення цієї мети дизайн веб-сайту повинен бути естетичним, функціональним та інформативним.

Успішний дизайн веб-сайту готелю повинен відповідати вимогам клієнтів та бути узгодженим з брендом готелю. Оптимізація веб-сайту для пошукових систем та адаптація для мобільних пристроїв також є важливими аспектами дизайну веб-сайту.

У процесі розробки дизайну веб-сайту готелю необхідно враховувати потреби та бажання клієнтів, щоб створити максимально зручний та ефективний інструмент для бронювання номерів та отримання інформації про готель. Правильно спроектований та розроблений дизайн веб-сайту готелю може забезпечити високий рівень задоволеності клієнтів та сприяти успішному розвитку бізнесу готелю [22].

Головна сторінка веб-сайту готелю є однією з ключових складових для привернення уваги потенційних клієнтів. На ній розташовані декілька найпопулярніших номерів, що сприяє залученню клієнтів та збільшенню продажу готельних послуг. Базова навігація дозволяє зручно орієнтуватися на сайті та знайти необхідну інформацію. Наявність можливості вибору дати в'їзду і виїзду, а

також кількості людей, сприяє вибору найбільш оптимальних номерів для проживання. Крім того, на головній сторінці можливо знайти інформацію про акції та спеціальні пропозиції, що також збільшує інтерес клієнтів до готелю (Рис. 2.1).

Таким чином, головна сторінка веб-сайту готелю виконує важливу роль у залученні та утриманні клієнтів, а також підвищенні конкурентоспроможності готелю на ринку готельних послуг.

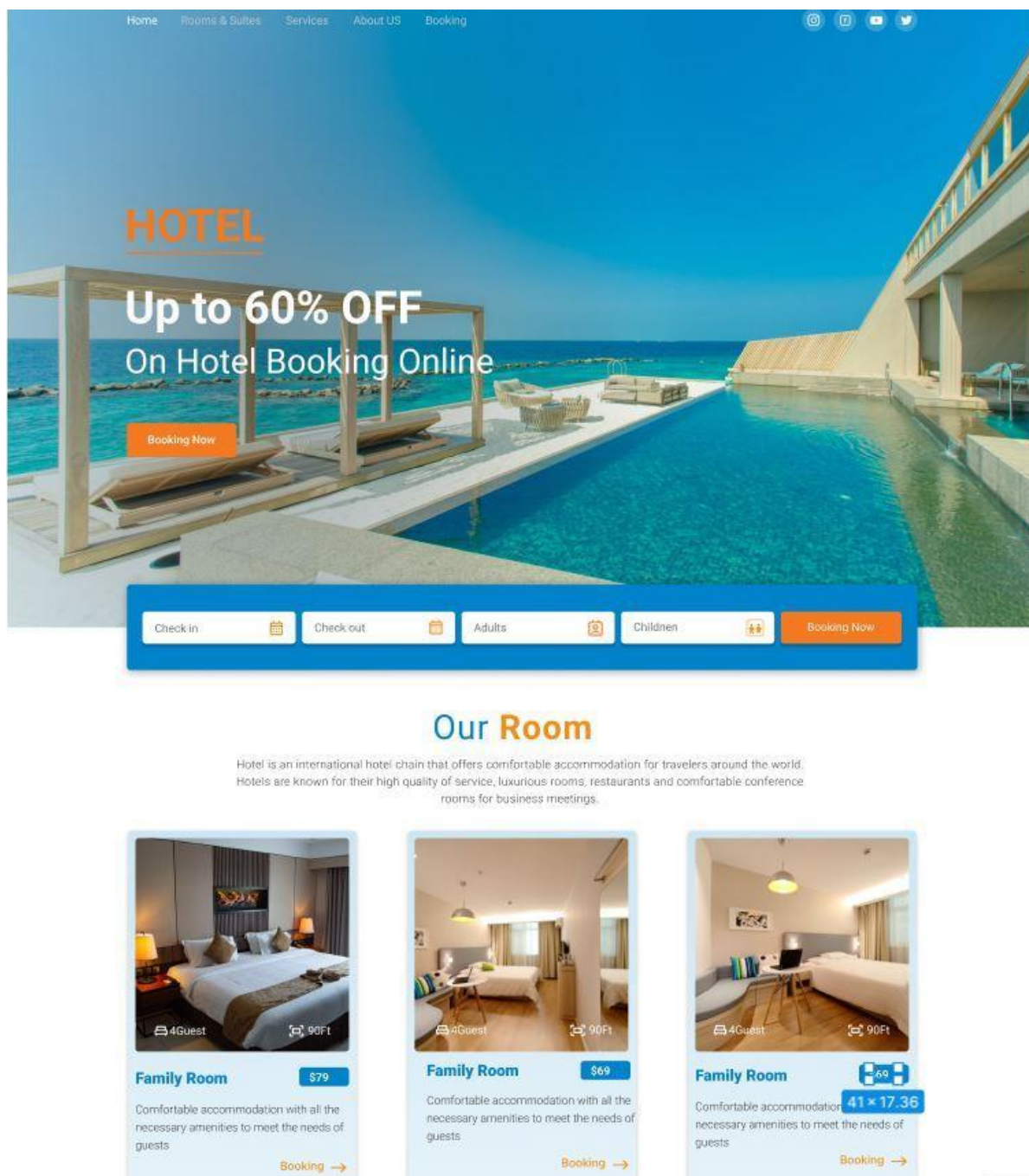


Рис 2.1 Головна сторінка веб-сайту готелю

Сторінка опису номера містить детальну інформацію про розташування номера, його площу, вид з вікна та інші важливі характеристики, а також фотографії номера з різних кутів, що надає можливість оцінити його комфортність та затишність. Крім того, на сторінці можна знайти ціну за добу проживання в

даному номері та можливість вибору дати в'їзду та виїзду, а також кількості осіб, які будуть проживати в номері, що дозволяє гостям з легкістю планувати свою подорож та вибрати найбільш відповідний для них варіант (Рис 2.2).

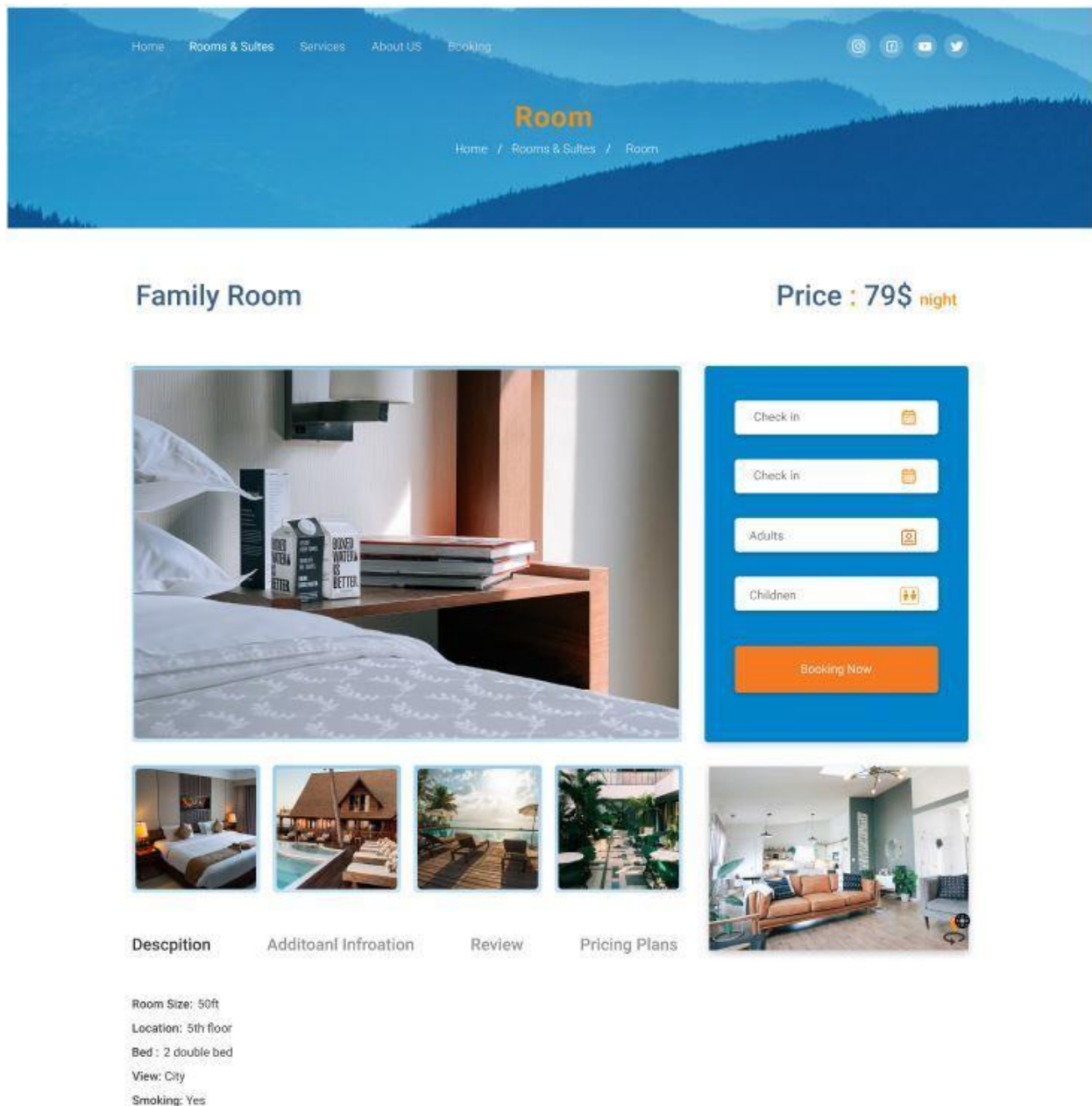


Рис. 2.2 Сторінка опису номера

На сторінці підтвердження замовлення, можна переглянути дату в'їзду і виїзду з номера, ціну та кількість спальних місць в номері (Рис. 2.2.3).

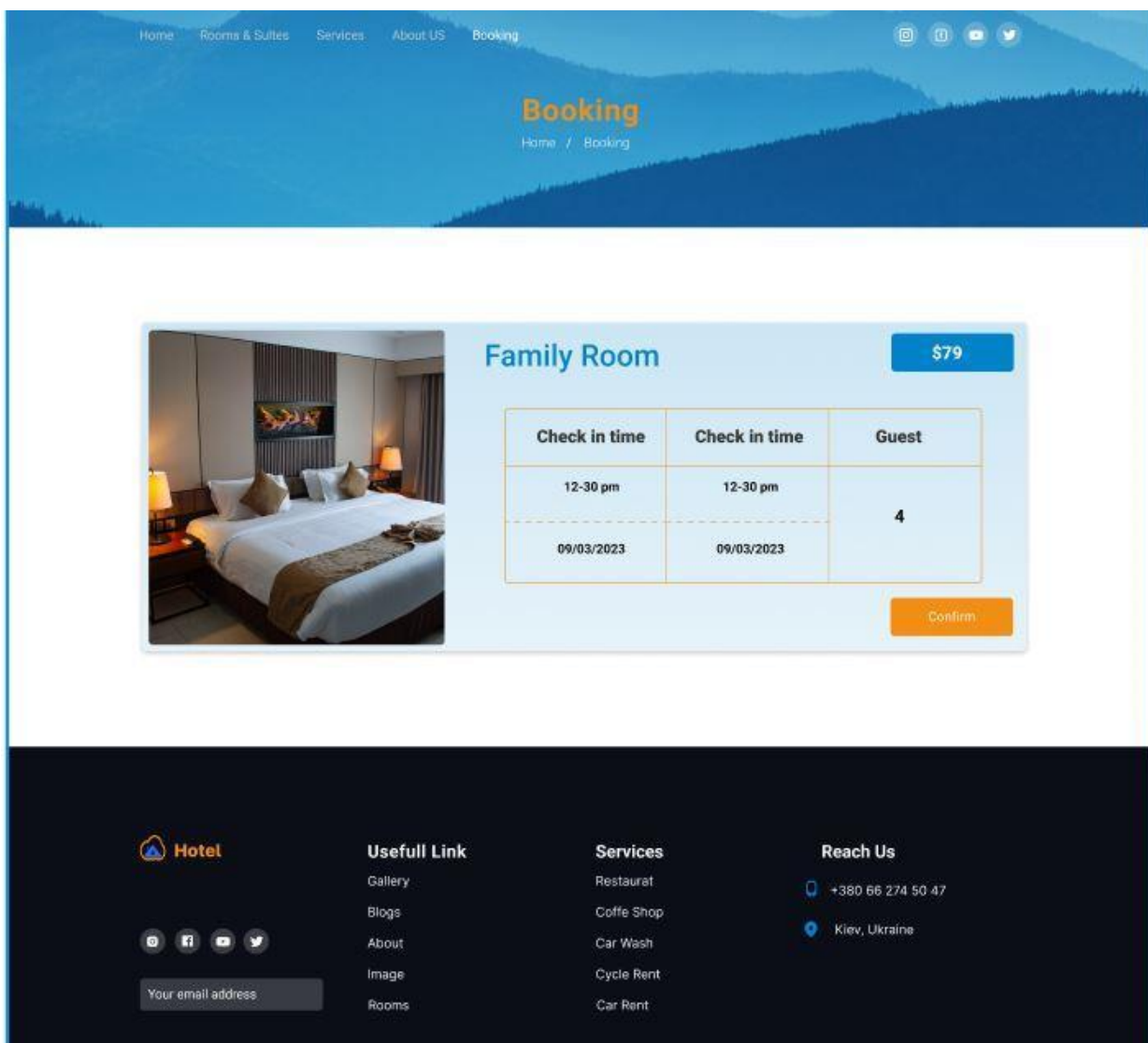


Рис. 2.3 Сторінка підтвердження замовлення

За допомогою дизайну веб-сайту можна створити неповторну атмосферу в готелі, яка допоможе залучити більше клієнтів і підвищити рівень задоволеності наявних. Важливо забезпечити зручний та логічний інтерфейс веб-сайту, щоб користувачі могли знайти необхідну інформацію про готель та здійснити бронювання легко та швидко. Крім того, дизайн повинен відображати унікальність готелю та залучати увагу до його особливостей, таких як комфортні номери,

розташування, обслуговування та інші. Оптимальне поєднання функціональності та дизайну є ключем до успіху веб-сайту готелю.

2.2. Архітектура та проєктування бази даних веб-сайту готелю

Архітектура веб-сайтів є процесом планування та проєктування технічних, функціональних та візуальних компонентів веб-сайту перед його розробкою, розробкою та розгортанням. Вона є важливим інструментом для дизайнерів веб-сайтів та розробників для розробки та створення веб-сайту.

Зазвичай архітектура вебдодатків складається з трьох основних компонентів: веббраузеру, вебсерверу та сервера бази даних. Веббраузер є ключовим компонентом, що взаємодіє з користувачем, отримує вхідні дані та керує логікою презентації, одночасно контролюючи взаємодію користувача з програмою. Вебсервер керує бізнес-логікою та обробляє запити користувачів, направляючи їх до потрібного компонента та керуючи всіма операціями програми. Сервер бази даних забезпечує необхідні дані для програми та виконує завдання, пов'язані з даними.

У цілому, архітектура веб-сайтів є важливою складовою процесу розробки веб-додатків, яка допомагає забезпечити ефективну та безперебійну роботу веб-сайту.

У традиційній 2-рівневій архітектурі відрізняються два компоненти: клієнтська система або інтерфейс користувача та серверна система, зазвичай якою є сервер бази даних. У цій архітектурі бізнес-логіка включена в інтерфейс користувача або сервер бази даних. Недоліком 2-рівневої архітектури є зниження продуктивності зі збільшенням кількості користувачів, а також проблеми безпеки, пов'язані з прямою взаємодією бази даних та пристроєм користувача. Серед додатків, які зазвичай будуються з використанням цієї архітектури, можна відзначити системи бронювання залізничних перевезень та системи керування контентом [1].

У 3-рівневій архітектурі виділяють три рівні: рівень презентації / рівень клієнта, рівень програми / рівень бізнесу та рівень даних. Рівень презентації відповідає за відображення інформації для користувача та взаємодію з ним. Рівень програми / рівень бізнесу містить бізнес-логіку додатка, яка обробляє запити користувачів та здійснює логіку додатка. Рівень даних містить базу даних та забезпечує зберігання та доступ до даних, які використовуються в додатку. 3-рівнева архітектура дозволяє забезпечити кращу продуктивність та безпеку додатка, а також дозволяє більш ефективно керувати процесами додатка.

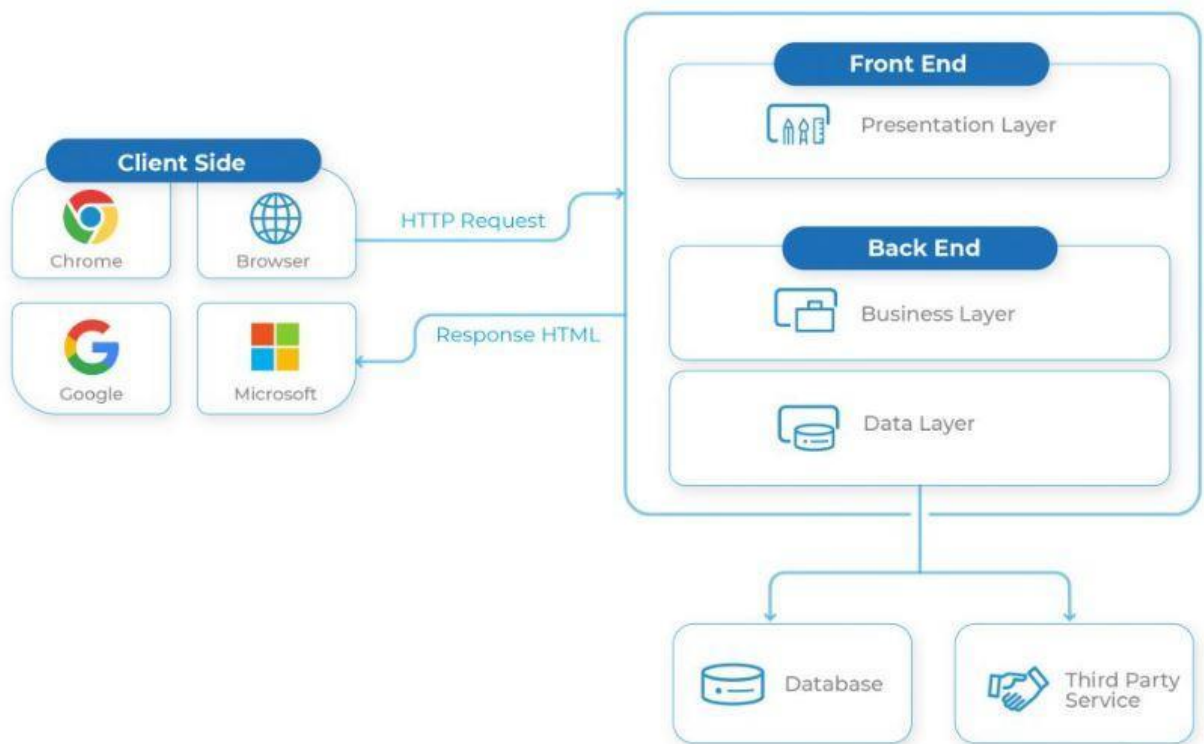


Рис. 2.4 Базова архітектура веб-сайту

В даній моделі, проміжні сервери приймають запити від клієнтів та обробляють їх, здійснюючи координацію з підлеглими серверами та використовуючи бізнес-логіку. Зв'язок між клієнтом та базою даних керується

проміжним прикладним рівнем, що дозволяє клієнтам отримувати доступ до даних з різних СУБД.

3-рівнева архітектура є більш безпечною, оскільки клієнт не має прямого доступу до даних. Можливість розгортання серверів додатків на кількох машинах забезпечує вищу масштабованість, кращу продуктивність та краще повторне використання. Вона може бути масштабована горизонтально, масштабуючи кожен елемент окремо [19]. Також можна абстрагувати основний бізнес від сервера бази даних, щоб ефективно виконувати балансування навантаження. Цілісність даних покращується, оскільки всі дані проходять через сервер додатків, який вирішує, як і хто має отримати доступ до даних. З цієї причини, зміна керівництва є легкою та економічно ефективною. Клієнтський рівень може бути тонким клієнтом, що означає зниження витрат на обладнання. Ця модульна модель дозволяє змінювати один рівень, не впливаючи на інші компоненти [21].

Архітектура вебдодатків включає дві ключові складові: клієнтський та серверний компоненти. Клієнтський компонент взаємодіє з серверним компонентом через браузер та забезпечує користувачам можливість отримання необхідної інформації. Він містить код, який відповідає за дизайн UI/UX, інформаційні панелі, сповіщення, налаштування конфігурації, макет та інтерактивні елементи.

Серверний компонент є ключовим компонентом архітектури вебдодатків. Він отримує запити користувачів, виконує бізнес-логіку та доставляє необхідні дані до зовнішніх систем. У складі серверного компонента містяться сервери, бази даних, вебсервіси та інші компоненти.

Інтерфейс прикладного програмування (API) є концепцією, яка дозволяє розробникам отримувати доступ до певних даних та функцій програмного забезпечення. Він виступає посередником, який дозволяє програмам спілкуватися один з одним та містить протоколи, інструменти та визначення підпрограм, необхідні для створення програм. Наприклад, коли користувач входить в

програму, програма викликає API, щоб отримати дані його облікового запису та облікові дані. Програма звернеться до відповідних серверів, щоб отримати цю інформацію та повернути її користувачеві. Веб-API – це API, доступний в Інтернеті через протокол HTTP та може бути розроблений з використанням таких технологій, як .NET і Java [29].

API є важливим інструментом для розробників програмного забезпечення, який дозволяє використовувати наявні функції замість їх повторної розробки, що призводить до зниження витрат на розробку продукту. Це, своєю чергою, допомагає збільшити продуктивність та підвищити швидкість виходу на ринок. API покращують ефективність розробки, забезпечуючи зручну інтерфейсну взаємодію між компонентами програмного забезпечення та допомагають зменшити витрати на розробку шляхом максимально можливого використання чинних ресурсів. Крім того, використання API покращує співпрацю та підключення в екосистемі, що дозволяє покращити взаємодію з клієнтами та забезпечити більш якісний сервіс.

Архітектура веб-сайту готелю містить базу даних PostgreSQL, бізнес-логіку бронювання номерів та вебклієнт, який включає головну сторінку, огляд кімнат та можливість бронювання номерів (Рис. 2.3.2).

База даних PostgreSQL забезпечує зберігання даних про кімнати готелю, ціни, доступність та іншу інформацію, яка необхідна для бронювання номерів. Ця база даних може бути інтегрована з системою керування готелем (Property Management System), щоб дозволити автоматичне оновлення інформації про наявність номерів та ціни.

Бізнес-логіка бронювання номерів може бути розроблена з використанням популярних фреймворків, і може забезпечувати автоматичне підтвердження бронювань, відправку електронної пошти клієнтам та оновлення бази даних.

Вебклієнт готелю може бути розроблений на основі HTML, CSS та JavaScript, щоб забезпечити користувачам зручний та привабливий інтерфейс

користувача. Головна сторінка може містити інформацію про готель, його послуги та переваги, а також може мати пошукову форму для пошуку вільних номерів на певну дату. Огляд кімнат може містити фотографії кімнат, їх опис та ціни. Можливість бронювання номерів може бути вбудована в вебклієнт, дозволяючи користувачам бронювати номери безпосередньо через сайт.

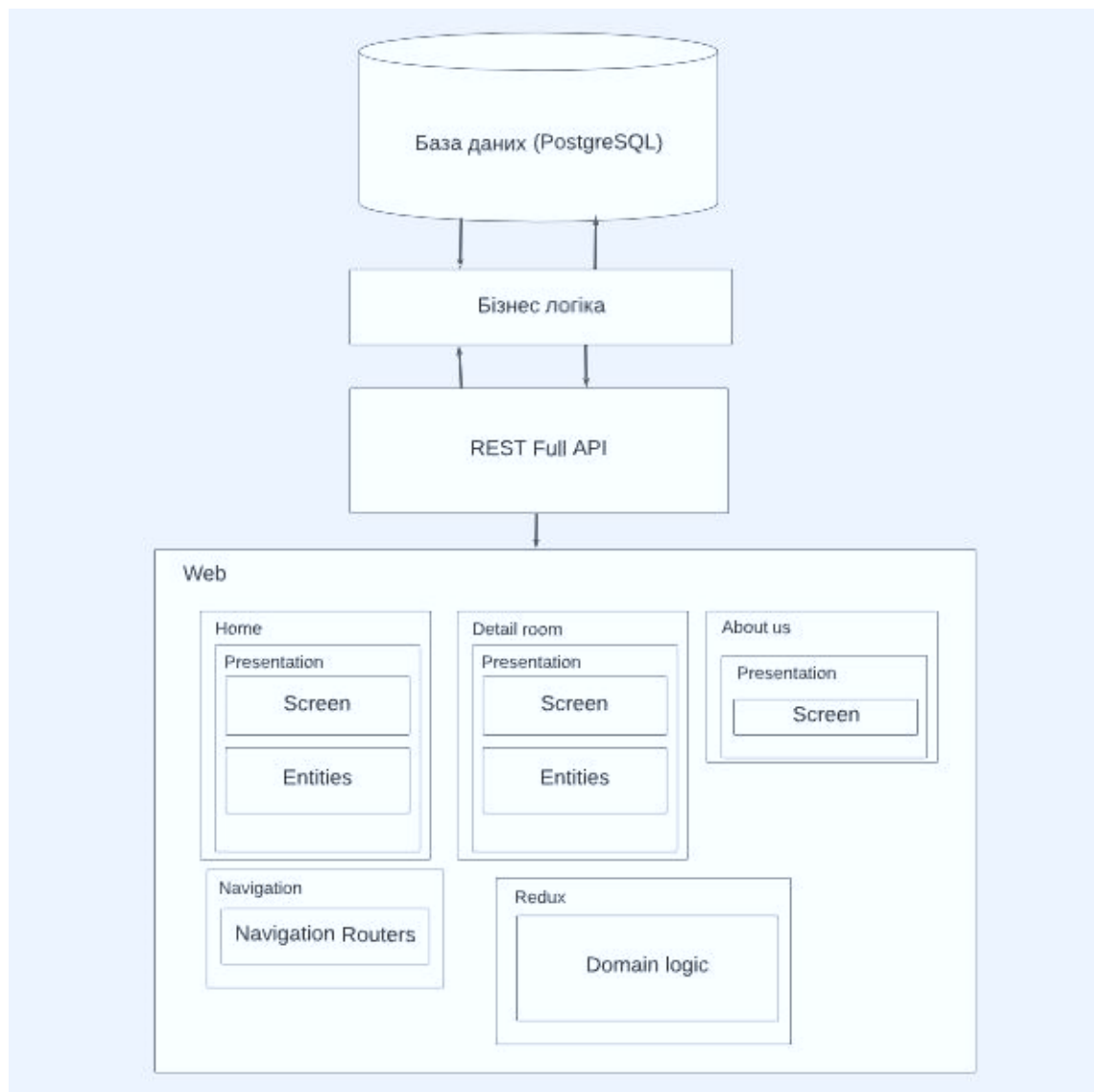


Рис. 2.5 Базова архітектура веб-сайту готеля

У сучасному світі програмного забезпечення, де існує висока конкуренція стосовно створення якісних продуктів та послуг, недостатньо мати лише якісний продукт. Важливість способу, яким продукт та послуги доставляються клієнтам, набуває все більшого значення. У цьому контексті, веб-додатки можуть служити ефективним засобом досягнення поставленої мети. Організації повинні створювати та використовувати високооптимізовані вебпрограми, які економічно ефективно забезпечують швидкість та продуктивність, а також надзвичайно привабливий інтерфейс користувача (UX). Розробка правильної архітектури вебпрограми є вирішальним чинником у досягненні цієї мети.

Для веб-сайту готелю взято реляційну базу даних. Реляційні бази даних забезпечують стабільність, надійність та швидкість операцій з даними, так як вони добре підходять для зберігання структурованої інформації в таблицях зі зв'язками між ними. Це дозволяє здійснювати різні операції з даними, такі як додавання, видалення, зміна та вибірка, з високою швидкістю.

Для веб-сайту готелю взято СУБД MySQL для зберігання даних про номери, бронювання, клієнтів та інші дані, які пов'язані з управлінням готелю.

MySQL підтримує багато типів даних, включаючи числа, рядки, дати та більш складні структури, такі як таблиці та стовпці. Вона також забезпечує високу швидкість операцій з даними, що дозволяє ефективно зберігати та отримувати інформацію з бази даних. Вона має відкритий вихідний код, що дозволяє розробникам створювати власні розширення та інструменти для покращення роботи з MySQL [30].

Провівши аналіз всіх об'єктів для веб-сайту готелю було розроблено схему бази даних (Рис. 2.6.).

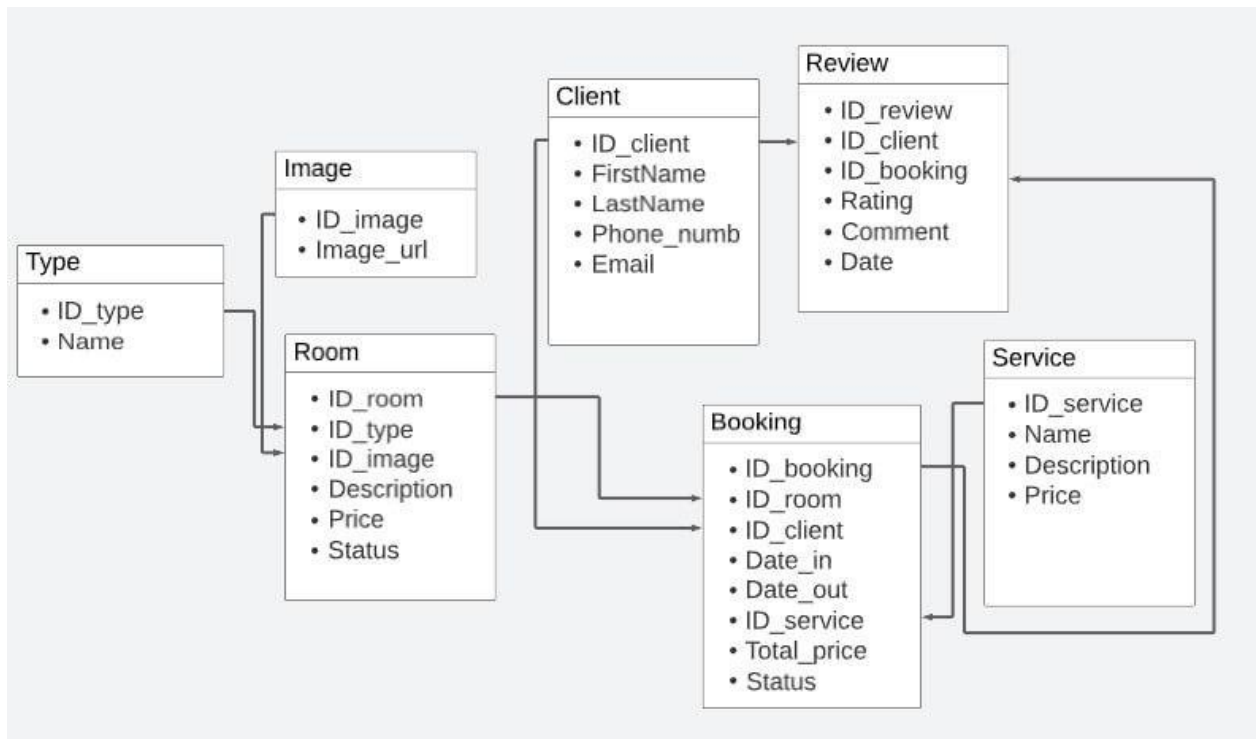


Рис. 2.6 Схема бази даних веб-сайту готеля.

Таблиця "Кімната" містить інформацію про номери готелю. Основні поля таблиці містять:

- ID номеру: унікальний ідентифікатор номеру.
- Тип номеру: тип номеру (одномісний, двомісний, люкс тощо).
- Опис: короткий опис номеру (наприклад, вид на море, балкон, ванна).
- Ціна: ціна за ніч за номер.
- Статус: показує, чи є вільний номер в готелі.
- Фотографії: зображення номеру.

Крім того, створюється зв'язок між таблицями "Кімната" та "Бронювання", щоб відслідковувати бронювання номерів клієнтами, за допомогою зв'язку "один до багатьох" (one-to-many), оскільки кожен номер може мати багато бронювань, а кожне бронювання може бути зроблене на один номер.

Таблиця "Кімната" допоможе відслідковувати інформацію про наявність номерів у готелі, їх тип та ціни, що дозволить клієнтам знайти відповідний номер за їхніми потребами та бюджетом.

Таблиця "Клієнти" містить інформацію про кожного клієнта готелю. Основна мета таблиці - зберігати особисту інформацію про клієнтів, щоб готель міг забезпечити кращий сервіс своїм клієнтам.

Основні поля таблиці "Клієнти" включають:

- ID клієнта: унікальний ідентифікатор, який дозволяє знайти інформацію про клієнта в базі даних.
- Прізвище та ім'я: особиста інформація про клієнта.
- Адреса електронної пошти: електронна адреса клієнта, за допомогою якої готель може зв'язатися з ним з приводу бронювання, підтвердження або інших питань.
- Номер телефону: номер телефону клієнта, що дозволяє готелю зв'язатися з ним для додаткової інформації або підтвердження бронювання.

Таблиця "Бронювання" зберігає інформацію про бронювання номерів готелю. Основні поля таблиці "Бронювання" включають:

- ID бронювання: унікальний ідентифікатор, який дозволяє знайти інформацію про бронювання в базі даних.
- ID клієнта: унікальний ідентифікатор клієнта, який здійснив бронювання.
- ID номеру: унікальний ідентифікатор номеру, який був заброньований.
- ID послуги: ідентифікатор послуги, які готель може надавати своїм клієнтам
- Дата початку бронювання: дата початку бронювання номеру.

- Дата кінця бронювання: дата кінця бронювання номеру.
- Статус бронювання: статус бронювання (наприклад, "заброньовано", "скасовано" або "підтверджено").
- Ціна: вартість бронювання номеру.

За допомогою цієї таблиці адміністратор готелю може відстежувати всі бронювання, включаючи кількість гостей, дату заїзду та виїзду, ціну та статус оплати. Також ця таблиця може використовуватися для відображення вільних та заброньованих номерів на веб-сайті готелю, щоб клієнти могли швидко та легко зробити бронювання.

Таблиця "Послуги" є важливою складовою бази даних готеля, оскільки вона містить інформацію про різні послуги, які готель може надавати своїм клієнтам. Готель може надавати різноманітні послуги, такі як харчування, розваги, транспорт та інші, і важливо мати детальну інформацію про ці послуги для їх ефективного маркетингу та продажу.

Основні поля таблиці "Послуги" включають:

- ID послуги: унікальний ідентифікатор послуги, який дозволяє знайти інформацію про послугу в базі даних.
- Назва послуги: назва послуги, яку надає готель.
- Опис послуги: детальний опис послуги, що надається готелем.
- Ціна: вартість послуги.

Таблиця відгуків (reviews) є однією з таблиць, яка може бути корисною для системи готелю. Ця таблиця міститиме інформацію про відгуки та оцінки клієнтів про свій досвід перебування в готелі. Нижче наведено список полів для цієї таблиці:

- id_review: унікальний ідентифікатор відгуку

- `id_booking`: ідентифікатор бронювання, до якого відноситься відгук
- `id_user`: ідентифікатор користувача, який залишив відгук
- `rating`: оцінка, яку користувач поставив для готелю (наприклад, від 1 до 5 зірок)
- `comment`: текстовий коментар користувача
- `date`: дата, коли був залишений відгук

За допомогою цієї таблиці інші користувачі можуть ознайомитися з думкою клієнтів про готель та прийняти рішення про бронювання.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ГОТЕЛЯ

Програмна реалізація веб-сайту готелю передбачає розробку функціональності сайту, яка буде взаємодіяти з базою даних готелю. Для цього потрібно використовувати певні технології та мови програмування.

Одним з основних компонентів веб-сайту готелю є інтерфейс користувача. Для розробки інтерфейсу можна використовувати HTML, CSS та JavaScript. HTML відповідає за структуру сторінки, CSS - за візуальний оформлення, а JavaScript - за інтерактивність.

Створення веб-сайту готелю слід розпочинати з написання HTML-коду для головної сторінки. Для головної сторінки за допомогою тегів створюється заголовок сторінки, додається зображення готелю та його логотипу, створюється навігаційне меню, де додається посилання на сторінки "Про нас", "Номери", "Бронювання" та "Контакти". Також додається форма пошуку номерів за датами, яку можна було б використовувати для бронювання номерів (Додаток А, Рис 3.1).

Для того, щоб створити головну сторінку HTML, використовують теги HTML, HEAD та BODY.

Тег HTML використовується для вказівки, що весь код сторінки HTML знаходиться всередині цього тегу.

HEAD використовується для включення інформації, яка не відображається на веб-сторінці, але все ж необхідна для правильного її відображення. В нашому випадку, включено мета-теги, які вказують на кодування символів і заголовок сторінки.

BODY використовується для вмісту, який відображається на сторінці. Створена структура сторінки, містить навігаційне меню, заголовок, зображення та інші необхідні елементи. Теги NAV та UL використовуються для створення навігаційного меню з посиланнями. Тег HEADER для вказання заголовку сторінки. Тег SECTION та ARTICLE для створення розділів та статей відповідно.

Тег IMG використовується для включення зображень на сторінці.. Інші HTML сторінки знаходяться в Додатку.

Після написання HTML-коду слід переходити до стилізації сторінки з допомогою CSS. Додано стилі для заголовків, тексту, зображень та навігаційного меню, щоб зробити сторінку більш привабливою та зручною для відвідувачів. Також створено адаптивний дизайн, щоб сторінка виглядала добре на різних пристроях та розмірах екрану (Додаток А, Рис 3.2).

CSS документ описує стилі для декількох елементів на головній сторінці:

- `.index-hero-img`: задає фонове зображення для головного слайдера, налаштовує його позиціонування, розмір та висоту.
- `.index-hero-box`: задає стиль для блоку з назвою готелю та кнопкою "Детальніше". Встановлює фоновий колір, тінь, відступи від країв та вирівнювання тексту.
- `.forest-img`: задає фонове зображення для блоку з фотографією, налаштовує його позиціонування, розмір та висоту. Встановлює фіксоване позиціонування, щоб зображення залишалося на місці під час прокручування сторінки.
- `.forest-img-box`: встановлює стиль для блоку з текстом на фоні зображення, встановлює фоновий колір, тінь, відступи від країв та вирівнювання тексту. Задає відступ зверху для розташування блоку з текстом посередині фотографії.
- `.carousel-item>img`: встановлює максимальну висоту зображень у слайдері.
- `.index-eat-img`: задає стиль для блоку з фотографією, встановлює параметр "overflow" для обрізання зайвих частин зображення.
- `.carousel-btn`: стиль для кнопки у слайдері, задає колір, тінь, розмір, мінімальну ширину та відступи від країв.

- `.carousel-btn: hover`: стиль для стану кнопки при наведенні на неї курсору миші.
- `.carousel-icons-style`: стиль для іконок-стрілок у слайдері, задає колір та розмір.

Після розробки стилізації CSS створено за допомогою JavaScript різноманітні ефекти та анімації на сторінці, а саме:

1. Карусель зображень: Використовуючи JavaScript створено карусель зображень, яка буде автоматично перемикає зображення певним інтервалом часу або перемикає зображення за допомогою кліків на кнопки.
2. Плавні ефекти при прокручуванні: Створено ефекти, що появляються або зникають під час прокручування сторінки, а саме ефект зникнення меню навігації, коли користувач прокручує сторінку вниз (Додаток А, Рис. 3.3).

Цей код дозволяє створити карусель зображень, яка містить кнопки "попередній слайд" та "наступний слайд", що можна викликати за допомогою кліку на відповідні кнопки. Крім того, код також включає плавні ефекти при прокручуванні, що змінюють зовнішній вигляд веб-сайту.

Для реалізації бронювання номеру створюється форма на сторінці сайту, яка дозволяє ввести дані гостя, такі як ім'я, прізвище, телефон та інші контактні дані, а також дати бронювання та номер кімнати. Після того, як користувач ввів всі необхідні дані, вони зберігаються в базі даних.

Для бронювання номера можна використовувати функцію `book_room`, яка приймає на вхід список номерів `rooms`, номер номера, дату початку бронювання та дату завершення бронювання. Функція перевіряє, чи доступний номер на вказаний період, і якщо так, встановлює для нього прапорець `is_booked` у значення `True` (Додаток А, Рис. 3.4).

Після реалізації функціоналу веб-сайту проводиться розгортання його на сервері. Один з найпростіших способів розгортання сайту - це використання хостинг-провайдера, який надає послуги хостингу та доменного імені. Для

розгортання сайту на хостинг-провайдері необхідно завантажити файли веб-додатку на сервер, налаштувати базу даних та встановити необхідні залежності. Після цього сайт можна перевірити, відкривши його в браузері за допомогою доменного імені [12].

Для розгортання веб-сайту готелю вибрано інструмент Docker. Docker - це інструмент, який дозволяє запускати програми в «віртуальному» середовищі-контейнері. Він дозволяє розгорнути веб-додатки та сервіси швидко та ефективно.

Щоб розгорнути веб-сайт за допомогою Docker, необхідно:

1. Написати Dockerfile, який описує веб-додаток та його залежності.
2. Збудувати Docker-образ за допомогою команди `docker build`.
3. Запустити контейнер зі збудованим Docker-образом за допомогою команди `docker run`.
4. Налаштувати веб-сервер або зворотній проксі, щоб прокинути трафік на запущений контейнер.

Після розгортання веб-сайту за допомогою Docker можна легко масштабувати веб-сайт, запускаючи більше контейнерів на більш потужному сервері або на кількох серверах в режимі кластеру.

Для початку роботи користувач має можливість з головної сторінки вибрати дату в'їзду та виїзду, вибрати кількість дорослих та дітей та натиснути кнопку `booking now`, для того щоб з'явилися вільні номери згідно параметрів (Рис. 3.7). Також, якщо не вказати дані, а просто натиснути на кнопку `booking now`, то користувач переходить на сторінку всіх доступних номерів для замовлення (Додаток А, Рис. 3.8).

Після того, як користувач вибрав номер на загальній сторінці всіх номерів, він переходить на детальну сторінку цього номеру з усіма фото цього номеру та детальним описом. Також на сторінці він має можливість вибрати дати, в які він буде перебувати в цьому номері та з якою кількістю дорослих та чи будуть в

номері діти (Додаток А, Рис. 3.9). Це робиться для того, щоб забронювати цей номер.

Також він має можливість вибрати додаткові послуги в готелі для комфортного відпочинку.

Після того, як користувач вже визначив, який він буде брати номер та визначився з датами, то він переходить на сторінку оформлення замовлення, де вводить основні дані про себе, а саме стать, прізвище, ім'я, телефон, електронну пошту (Додаток А, Рис.3.10).

В кінці користувач переходить на останню сторінку підтвердження бронювання, де вказується інформація та сума бронювання (Додаток А, Рис. 3.11). Користувач має змогу оплатити зараз, або оплатити при прибутті в готель.

Веб-сайт готелю надає користувачам зручну можливість бронювання номерів та вибору додаткових послуг онлайн. Користувачі можуть з легкістю знайти вільні номери за певними параметрами, переглянути детальну інформацію про кожний номер та забронювати його за допомогою простої та зрозумілої форми. Остання сторінка підтвердження дозволяє користувачеві переглянути своє бронювання та вибрати зручний спосіб оплати. Все це робить процес бронювання готелю більш зручним та швидким для користувачів, а також дозволяє готелю ефективніше керувати своїми ресурсами та покращити якість обслуговування своїх клієнтів.

Сайт кваліфікаційної роботи можна переглянути за посиланням:

https://dmytromeleshko2.github.io/Hotel_Site/

ВИСНОВКИ

Під час виконання кваліфікаційного проєкту було розроблено веб-сайт готелю. Для початку було проаналізовано актуальність розробки та сучасний стан предметної області в Україні. Розробка веб-сайту для готелю є дуже актуальним завданням в сучасних умовах, оскільки Інтернет став основним джерелом інформації для багатьох людей. Більшість туристів планують свої подорожі та бронюють готелі через Інтернет. Отже веб-сайт може бути дуже важливим інструментом для готелю, щоб привернути нових клієнтів та збільшити прибуток.

Стан готельної справи в Україні знаходиться в процесі розвитку. Однак, він відстає від розвинених країн Європи та світу. Україна має великий потенціал для розвитку готельної справи, оскільки в країні багато туристичних об'єктів та красивих місць для відпочинку. Тому, розробка веб-сайту для готелю може допомогти привернути більше туристів та зробити українську готельну справу більш конкурентоспроможною на світовому ринку.

Далі було проаналізовано сучасні інструменти розробки веб-сайту та основні етапи розробки. Сучасні інструменти розробки веб-сайтів дозволяють швидко і ефективно створювати функціональні та естетично привабливі веб-сайти готелів. До них входять фреймворки та бібліотеки, такі як React, які дозволяють створювати ефективні клієнтські додатки з високою продуктивністю та гнучкістю.

Етапи створення веб-сайту готелю включають планування, дизайн, розробку. Планування полягає в аналізі вимог клієнтів та визначенні функціональних можливостей веб-сайту. Дизайн включає розробку інтерфейсу користувача та візуального оформлення. Розробка полягає в програмуванні та розробці бази даних, а також в налаштуванні веб-сервера та розгортання веб-сайту.

Потім створено ефективну архітектуру сайту, створено базу даних та реалізовано сам веб-сайт готелю. База даних містить необхідну інформацію про

клієнтів, номери та бронювання, що дозволяє користувачам швидко та зручно здійснювати бронювання. Веб-сайт готелю має зрозумілу структуру та інтуїтивно зрозумілий інтерфейс, що сприяє залученню нових клієнтів та збереженню старих. Також реалізація функціоналу, такого як вибір номерів, внесення змін до бронювання та оплата відповідає сучасним стандартам веб-розробки.

Для забезпечення зручності користування веб-сайтом необхідно було створити інтуїтивно зрозумілий та логічний інтерфейс. Користувач повинен мати змогу з легкістю знайти необхідну інформацію, вибрати номер, вибрати дати проживання та додаткові послуги. Крім того, було важливо забезпечити легку взаємодію з платіжними системами та сервісами доставки.

Отже, в ході виконання кваліфікаційної роботи було отримано практичні навички створення веб-сайту готелю. А також, досвід та компетенції, які можуть бути використані в подальшій професійній діяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Архітектура клієнт-сервер. Електронна бібліотека – ВСП «Павлоградський фаховий коледж НТУ «Дніпровська політехніка». URL: <http://inter.ptngu.com/> (дата звернення: 12.04.2023).
2. Бліщук К. М., Козак І. І. Брендинг у сфері готельного бізнесу. *Efficiency of public administration*. 2022. № 68/69. С. 22–32. URL: <https://doi.org/10.36930/506802> (дата звернення: 08.03.2023).
3. Бронювання готелів – Вікіпедія. *Вікіпедія*. URL: http://uk.wikipedia.org/wiki/Бронювання_готелів (дата звернення: 05.03.2023).
4. День туризму 2022 разом з Visit Ukraine. *Visit Ukraine - RULES OF SAVE VISIT TO UKRAINE*. URL: <https://visitukraine.today/uk/blog/927/den-turizma-2022-vmeste-s-visit-ukraine> (дата звернення: 08.03.2023).
5. Звіт за результатами опитування "Проведення дослідження внутрішнього та виїзного туризму українців". URL: <https://www.tourism.gov.ua/statistic> (дата звернення: 12.03.2023).
6. Інноваційні підходи до моделювання бізнес-процесів підприємств готельно-ресторанного господарства / С. Ю. Попова та ін. *Вісник Хмельницького національного університету*, м. Хмельницький, 27 трав. 2020 р. Донецьк, 2020. С. 151–153.
7. Кобюк С. І. Розвиток готельного бізнесу в Україні : thesis. 2019. URL: <https://er.knutd.edu.ua/handle/123456789/14157> (дата звернення: 28.03.2023).
8. Корчак М., Рудь А., Грушецький С. Активізація процесу навчання при викладанні дисципліни «Система технологій в галузі механізації та електрифікації сільського господарства» спеціальності «Фінанси, банківська справа та страхування». *Сучасні тенденції забезпечення якості підготовки фахівців: проблеми та шляхи їх вирішення в умовах глобалізації та євроекономічної*

інтеграції. 2022. С. 358–364. URL: <https://isg-journal.com/isjel/article/view/37> (дата звернення: 18.03.2023).

9. Лавренова О., Стрельников А., Могил С. Управління якістю надання адміністративних послуг. Нац. ун-т "Од. юрид. акад.", 2019. URL: <https://doi.org/10.32837/11300.12140> (дата звернення: 02.05.2023).

10. Марусей Т. Пріоритети розвитку digital-маркетингу готельного бізнесу. *Економіка та суспільство*. 2021. № 26. URL: <https://doi.org/10.32782/2524-0072/2021-26-76> (дата звернення: 18.04.2023).

11. Новак К. А. Поняття про бізнес-процес. Етапи та методи математичного моделювання бізнес-процесів. Київ, 2017. 20 с.

12. Перенесення сайту на новий хостинг і зберегти його роботу: покрокове керівництво. *Inweb*. URL: <https://inweb.ua/blog/ua/kak-perenesti-sayt-na-novyy-hosting-i-sohranit-ego-rabotu-poshagovoe-rukovodstvo/> (дата звернення: 01.05.2023).

13. Піскозуб Л. Й. Структура рекламного контенту веб-сайтів. *Поліграфія і видавнича справа*. 2017. № 1 (73). С. 209–213.

14. Поцелуй А. А. Дослідження систем захисту веб-сайтів : thesis. 2017. URL: <http://elartu.tntu.edu.ua/handle/123456789/19396> (дата звернення: 22.03.2023).

15. Про туризм : Закон України від 15.09.1995 р. № 324/95-ВР : станом на 1 квіт. 2023 р. URL: <https://zakon.rada.gov.ua/laws/show/324/95-vr#Text> (дата звернення: 11.03.2023).

16. Управління бізнес-процесами в ГРС / В. Луцяк та ін. Вінниця : МОН України вінн. нац. аграр. ун-т, 2019. 331 с.

17. Федорова М. В., Білоцерківський О. Б. Міжнародний туризм як важлива галузь світової економіки : thesis. 2018. URL: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/41579> (дата звернення: 08.05.2023).

18. Шиш М. С. Веб-сайт туристичної агенції: thesis. 2021. URL: <http://ir.stu.cn.ua/123456789/22800> (дата звернення: 21.04.2023).
19. Що таке архітектура веб-сайтів? - визначення з техопедії - Розвиток 2023. *Icy Science*. URL: <https://uk.theastrologypage.com/website-architecture#:~:text=Архітектура%20веб-сайтів%20-%20це%20планування,розробки%20та%20розробки%20веб-сайту> (дата звернення: 14.04.2023).
20. Antonova I. Market analysis of hotel services in ukraine. *Baltic journal of economic studies*. 2015. P. 10–16. URL: <https://doi.org/10.30525/2256-0742/2015-1-1-10-16> (date of access: 03.04.2023).
21. Banga S. What is web application architecture? Components, models, and types. *Hackr.io*. URL: <https://hackr.io/blog/web-application-architecture-definition-models-types-and-more> (date of access: 21.04.2023).
22. Booking.com: the largest selection of hotels, homes, and holiday rentals. *Booking.com*. URL: <https://medium.com/keycafe/the-history-of-booking-com-cc213a735380> (date of access: 21.04.2023).
23. Chumak I. A., Reutskova O. M. Hotel business in ukraine: trends and prospects. *WEB-ресурс науково-практичних конференцій | Наукові, науково-практичні конференції та публікації, новини науки*. URL: http://confcontact.com/2013_04_04_dnu/9_Chumak.htm (date of access: 15.04.2023).
24. Gadin E. Cross platform compatibility for native Android hardware with React Native and React Native Web: thesis. 2021. URL: <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-453803> (date of access: 15.04.2023).
25. Herbert D. What is react.js? (uses, examples, & more). *HubSpot Blog | Marketing, Sales, Agency, and Customer Success Content*. URL: <https://blog.hubspot.com/website/react-js> (date of access: 03.04.2023).

26. Povorozniuk I. M., Chirva O. G., Maliarchuk N. M. Problems of small hotel business development in Ukraine. *Economies' horizons*. 2021. No. 1(12). P. 43–49. URL <http://reposit.nupp.edu.ua/handle/PoltNTU/4133> (date of access: 06.04.2023).

27. Starwebdesign com ua - Just another WordPress site. *Starwebdesign com ua - Just another WordPress site*. URL: <http://starwebdesign.com.ua/> (date of access: 03.04.2023).

28. The 10 best website design best practices. URL: <https://www.upwork.com/resources/website-design-best-practices> (date of access: 01.04.2023).

29. Web application architecture: the latest guide 2022. *ClickIT*. URL: <https://www.clickittech.com/devops/web-application-architecture/#:~:text=A%20web%20app%20architecture%20presents,the%20backend%20server%20can%20understand>. (date of access: 14.04.2023).

30. Web design best practices to attract more website visitors (2023). *Hostinger Tutorials*. URL: <https://www.hostinger.com/tutorials/web-design-best-practices> (date of access: 13.04.2023).

31. Website development: 8 steps + tips | mailchimp. *Mailchimp*. URL: <https://mailchimp.com/resources/guide-to-website-development/> (date of access: 12.03.2023).

32. Website development process: a step-by-step guide. *Chicago Digital Marketing & Web Development Agency | Marcel Digital*. URL: <https://www.marceldigital.com/blog/website-development-process-a-step-by-step-guide> (date of access: 08.04.2023).

33. Web technology for developers | MDN. *MDN Web Docs*. URL: <https://developer.mozilla.org/en-US/docs/Web> (date of access: 12.04.2023).

34. What is HTML? Hypertext markup language basics explained. *Hostinger Tutorials*. URL: <https://www.hostinger.com/tutorials/what-is-html> (date of access: 18.03.2023).

35. What is javascript? A basic introduction to JS for beginners. *Hostinger Tutorials*. URL: <https://www.hostinger.com/tutorials/what-is-javascript> (date of access: 08.05.2023).

36. What is CSS, how does it work and what is it used for? - skillcrush. *Skillcrush*. URL: <https://skillcrush.com/blog/css/> (date of access: 14.04.2023).

37. What is Lucidchart learning center. *Lucidchart*. URL: <https://www.lucidchart.com/pages/tour> (date of access: 22.04.2023).

38. What is web design? The ultimate guide to website design [2020] - pagecloud blog - web design & small business tips for your website. *Pagecloud.com*. URL: <https://www.pagecloud.com/blog/web-design-guide> (date of access: 14.04.2023).

39. Zalevska O., Ladogubets T., Lavro O. Using drag-and-drop technology when creating a web interface. *Modern problems of modeling*. 2020. Vol. 19. P. 57–64. URL: <https://doi.org/10.33842/2313-125x/2020/19/57/64> (date of access: 02.05.2023).

40. Zemlina Y., Lifirenko O. Trends in business hotel development in Ukraine. *Restaurant and hotel consulting. innovations*. 2019. Vol. 2, no. 1. P. 121–131. URL: <https://doi.org/10.31866/2616-7468.2.1.2019.170430> (date of access: 28.03.2023).

ДОДАТКИ

Додаток А

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Hotel</title>
6  </head>
7  <body>
8      <header>
9          <nav>
10             <ul>
11                 <li><a href="#">Головна</a></li>
12                 <li><a href="#">Номери</a></li>
13                 <li><a href="#">Бронювання</a></li>
14                 <li><a href="#">Про нас</a></li>
15                 <li><a href="#">Контакти</a></li>
16             </ul>
17          </nav>
18      </header>
19      <main>
20          <section>
21              <h1>Ласкаво просимо в готель!</h1>
22              <p>Ласкаво просимо до нашого готелю, розташованого в самому серці Києва з чудовим видом на річку Дніпро. Наш сучасними зручностями, щоб забезпечити незабутній відпочинок для наших гостей. Крім того, наш готель має рест чудовим видом на річку під час сніданку, обіду або вечері. Наші привітні та професійні співробітники завжди п максимально комфортним та приємним.</p>
23          </section>
24          <section>
25              <h2>Номери</h2>
26              <p>Наш готель пропонує комфортабельні номери різної категорії, щоб задовольнити потреби різних типів гостей. обладнаний сучасними зручностями та меблями, щоб зробити ваше перебування найбільш комфортним. В нашому готел крісла, ванну кімнату з душем, фен, туалетні засоби та інші зручності.</p>
27          </section>
28          <section>
29              <h2>Бронювання</h2>
30              <p>Щоб забронювати номер в нашому готелі, Вам необхідно обрати відповідний номер з доступних пропозицій, а п менеджери з радістю допоможуть вам вирішити всі питання та організувати бронювання номеру. Після підтвердженн включаючи дати, тип номеру, вартість та інші важливі відомості.</p>
31          </section>
32          <section>
33              <h2>Про нас</h2>
34          </section>
35          <section>
36              <h2>Контакти</h2>
37              <p>Контактні дані готелю</p>
38          </section>
39      </main>
40      <footer>
41          <p>Copyright ©
42          <script>
43              document.write(new Date().getFullYear())
44          </script>
45          Hotel</p>
46      </footer>
47  </body>
48  </html>

```

Рис. 3.1 Головна сторінка HTML документа.

```

1  .index-hero-img {
2      background: url('/static/images/index_hero_image.jpg') no-repeat center;
3      background-size: cover;
4      width: 100%;
5      min-height: 800px;
6  }
7
8  .index-hero-box {
9      min-height: 70px;
10     background-color: #fff;
11     box-shadow: 0 0 2px 2px #D3E0E7;
12     padding-left: 0px;
13 }
14
15 .index-hero-box h1 {
16     color: #251D1D !important;
17 }
18
19 .forest-img {
20     background: url('/static/images/index_forest_image.jpg') no-repeat center;
21     background-size: cover;
22     width: 100%;
23     min-height: 800px;
24     background-attachment: fixed;
25 }
26
27 .forest-img-box {
28     background-color: #FFF;
29     box-shadow: 0 10px 10px 0 #251D1D;
30     margin-top: 15%;
31     border: 1px solid #466060;
32 }
33
34 .carousel-item>img{
35     max-height: 700px;
36 }
37
38 .index-eat-img {
39     overflow: hidden;
40 }
41
42 .carousel-btn {
43     border: 2px solid #251D1D;
44     color: #251D1D;
45     background-color: #fff;
46     font-size: 30px;
47     margin-bottom: 280px;
48     min-width: 300px;
49 }
50
51 .carousel-btn:hover {
52     color: #fff;
53     background-color: #251D1D;
54 }
55
56 .carousel-icons-style {
57     color: #76A1B1;
58     font-size: 40px;
59 }

```

Рис. 3.2 Головна сторінка CSS документу

```

1  const carousel = document.querySelector('.carousel');
2  const inner = document.querySelector('.carousel-inner');
3  const items = document.querySelectorAll('.carousel-item');
4  const prevBtn = document.querySelector('.prev');
5  const nextBtn = document.querySelector('.next');
6
7  let currentSlide = 0;
8
9  function moveToSlide(slide) {
10     inner.style.transform = `translateX(-${slide * items[0].clientWidth}px)`;
11 }
12
13 prevBtn.addEventListener('click', () => {
14     currentSlide--;
15     if (currentSlide < 0) {
16         currentSlide = items.length - 1;
17     }
18     moveToSlide(currentSlide);
19 });
20
21 nextBtn.addEventListener('click', () => {
22     currentSlide++;
23     if (currentSlide >= items.length) {
24         currentSlide = 0;
25     }
26     moveToSlide(currentSlide);
27 });
28
29 let prevScrollpos = window.pageYOffset;
30 window.onscroll = function() {
31     let currentScrollpos = window.pageYOffset;
32     if (prevScrollpos > currentScrollpos) {
33         document.querySelector("header").style.top = "0";
34     } else {
35         document.querySelector("header").style.top = "-80px";
36     }
37     prevScrollpos = currentScrollpos;
38 }
39
40

```

Рис. 3.3 JS документ для головної сторінки веб-сайту

```

1  from datetime import datetime, timedelta
2
3  class Room:
4      def __init__(self, room_number, capacity, price, is_booked=False):
5          self.room_number = room_number
6          self.capacity = capacity
7          self.price = price
8          self.is_booked = is_booked
9
10     room_101 = Room(101, 2, 1000)
11     room_102 = Room(102, 1, 800)
12     room_103 = Room(103, 4, 1500)
13     room_104 = Room(104, 2, 1100)
14
15     rooms = [room_101, room_102, room_103, room_104]
16
17     def book_room(rooms, room_number, start_date, end_date):
18         for room in rooms:
19             if room.room_number == room_number and not room.is_booked:
20                 start_date = datetime.strptime(start_date, '%Y-%m-%d')
21                 end_date = datetime.strptime(end_date, '%Y-%m-%d')
22                 delta = end_date - start_date
23                 for i in range(delta.days):
24                     date = start_date + timedelta(days=i)
25                     if date.strftime('%Y-%m-%d') in room.bookings:
26                         return False
27                 room.is_booked = True
28                 return True
29         return False

```

Рис. 3.4 Фрагмент коду для бронювання номеру

```

1 FROM python:3.10.10-slim-buster
2
3 RUN mkdir "/hotel"
4 WORKDIR /hotel
5 COPY requirements.txt .
6
7 RUN pip install -r requirements.txt
8
9 COPY ...
10
11 version: "3.9"
12 services:
13   app:
14     container_name: hotel_app
15     build:
16       context: .
17     restart: always
18     ports:
19       - "8001:8000"
20     volumes:
21       - ./data/envs:/eva/data/envs
22       - ./data/static:/eva/data/static
23       - ./data/migrations:/eva/data/migrations
24     depends_on:
25       - db
26     networks:
27       - app-network
28       - nginx_proxy-network
29     command: /bin/sh -c "./app.sh ./data/envs/.env web"
30
31   db:
32     container_name: hotel_postgres
33     image: postgres:15.2-alpine
34     restart: always
35     ports:
36       - "5499:5432"
37     env_file:
38       - ./data/envs/postgres.env
39     volumes:
40       - ./data/postgres/data:/var/lib/postgresql/data
41     networks:
42       - app-network
43
44 networks:
45   app-network:
46     driver: bridge
47   nginx_proxy-network:
48     external: true
49

```

Рис. 3.5 Розгортання сайту за допомогою Docker

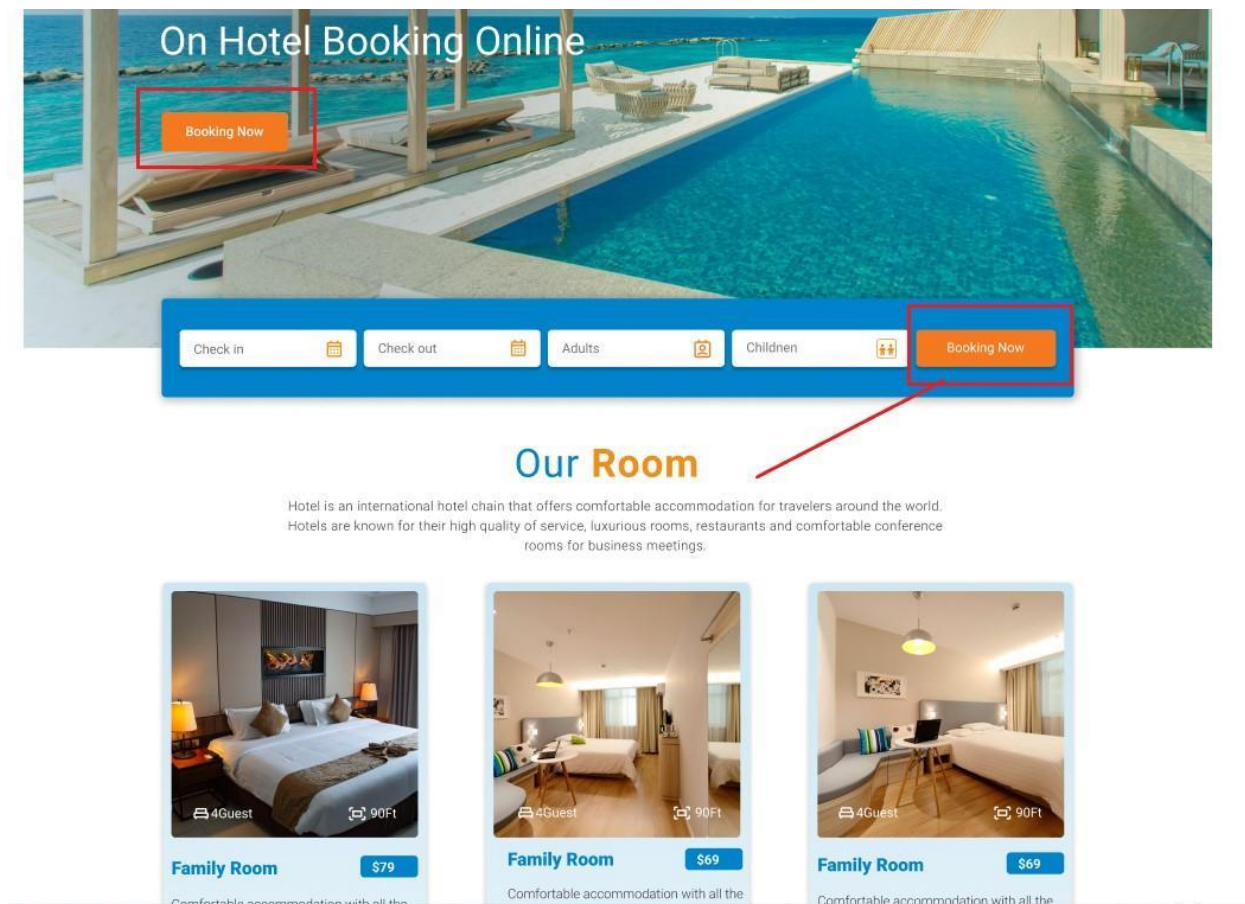


Рис. 3.6 Головна сторінка

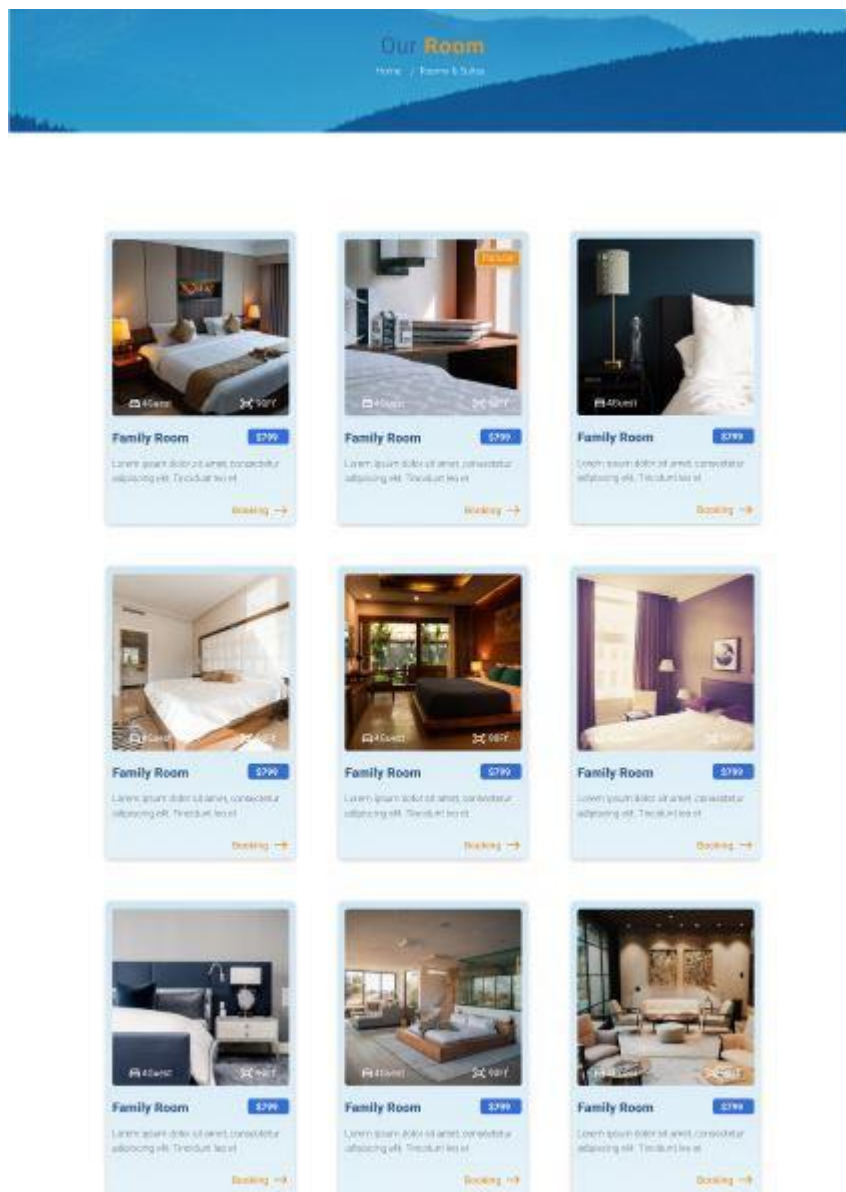


Рис. 3.8 Сторінка всіх номерів

AVAILABILITY

August						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

September						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

RESERVATION FORM

	CHECK-IN DATE
	CHECK-OUT DATE
	NO. of ADULTS
	NO. of CHILDREN

Checkout

Рис. 3.9 Форма для бронювання номеру

Your data

Information for booking

☒ I book for myself

☐ I am booking for another guest

Name

Name

E-mail

Country code and phone number

Number #1

Estimated time of arrival (optional)

Standard check-in time 14:00

Estimated time of departure (optional)

Standard check-out time 12:00

Рис. 3.10. Форма для заповнення даних користувача

Your reservation

CITYHOTEL
Standard Double

 Adults: 1

 Children: 0

 Arrival: 2023-05-08 14:00

 Departure: 2023-05-09 12:00

PROMO

No breakfast

Price per room	EUR 67.5
No taxes	EUR 63.08
Taxes	EUR 4.42
Meeting	EUR 0

EVERYTHING
EUR 67.5

Рис. 3.11. Форма підтвердження бронювання

Додаток Б

```

{% extends "base.html" %}
{% load static %}
{% block extra_css %}
    <link rel="stylesheet" href="{% static 'about/css/about.css' %}">
{% endblock %}
<!--Sets this template as current page in the menu-->
{% block nav_about %} active-menu {% endblock %}
<!--Adds extra title for current page-->

```

```

{% block extra_title %}- About{% endblock %}
{% block content %}
    <!--Hero image section-->
    <div class="container-fluid about-hero-img">
        <div class="row">
            <div class="col d-flex justify-content-end text-sm-left pr-0 pl-0">
                <div class="top-title">
                    <h1 class="standard-title">About</h1>
                </div>
            </div>
        </div>
    </div>
</div>
<!--Introduction section-->
<div class="container-fluid">
    <div class="row standard-header-box">
        <div class="col-12">
            <h1 class="standard-title box-title text-center">Your stay, your way.</h1>
        </div>
    </div>
    <div class="row">
        <div class="col-12 col-lg-6 ml-lg-0 introduction-bg">
            <h2 class="standard-title mt-5" data-aos="fade-up" data-aos-
duration="2000">Morning hike or a cup of tea?</h2>
            <p class="standard-text mt-5 mb-5 mr-5" data-aos="fade-up" data-aos-
duration="2000">We believe that the best experiences are those that
                we create ourselves. We've therefore built our whole hotel concept with you
                in mind. From the freedom to let you design your own retreat, to the locally
                sourced

```

```

        ingredients used in our kitchen and to our careful choice of organic products
        used in our spa. Our attention to detail means that you can focus on what
        matters most to you: creating a retreat to meet your specific needs.</p>
    </div>
    <div class="col-12 col-lg-6 organic-img">
    </div>
</div>
</div>
<!--Reviews section-->
<div class="container-fluid">
    <div class="row standard-header-box">
        <div class="col-12">
            <h1 class="standard-title box-title text-center">What our guests have said
about us.</h1>
        </div>
    </div>
    <div class="row">
        {% if not reviews %}
            {% include 'about/includes/no-reviews.html' %}
        {% else %}
            {% include 'about/includes/user-reviews.html' %}
        {% endif %}
        {% include 'about/includes/new-review-form.html' %}
    </div>
</div>
<!--Steps section-->
<div class="container-fluid">
    <div class="row standard-header-box mt-5">

```

```

    <div class="col-12">
        <h1 class="standard-title box-title text-center">Get started with three
steps.</h1>
    </div>
</div>
{% include 'about/includes/booking-steps.html' %}
</div>
<!--Links section-->
<div class="container-fluid">
    <div class="row standard-header-box">
        <div class="col-12">
            <h1 class="standard-title box-title text-center">Start browsing now.</h1>
        </div>
    </div>
</div>
<div class="container-fluid links-img">
    <div class="row center-align text-center">
        <div class="col-12 col-lg-4 mt-lg-5">
            <a class="btn standard-btn dark-btn" href="{% url 'all_stay_services'
%}">Stay</a>
        </div>
        <div class="col-12 col-lg-4 mt-lg-5">
            <a class="btn standard-btn dark-btn" href="{% url 'all_relax_services'
%}">Relax</a>
        </div>
        <div class="col-12 col-lg-4 mt-lg-5">
            <a class="btn standard-btn dark-btn" href="{% url 'all_eat_services'
%}">Eat</a>

```

```

        </div>
    </div>
</div>
{% endblock %}
.about-hero-img {
    background: url('/static/images/about_hero_image.jpg') no-repeat center;
    background-size: cover;
    width: 100%;
    min-height: 600px;
    background-attachment: fixed;
}
/*----- Steps section */
.step-title {
    font-size: 30px;
    font-family: 'Poiret One', cursive;
    color: #6494A6;
}
.organic-img {
    background: url('/static/images/about_organic.jpg');
    width: 100%;
    max-height: 900px;
    background-repeat: no-repeat;
    background-size: cover;
    background-position-y: 80%;
}
.introduction-bg {
    background-color: #fff;
    padding-left: 50px;

```

```

}
.step-title {
    font-size: 100px;
}
/*----- Reviews section */
.service-thumbnail {
    width: 220px;
    height: 100px;
    object-fit: cover;
    border: 2px solid #251d25;
    margin-bottom: 30px;
}
.review-subtitle p {
    margin-bottom: 0px;
}
/*----- Links section */
.links-img {
    background: url('/static/images/index_forest_image.jpg') no-repeat center;
    background-size: cover;
    width: 100%;
    min-height: 400px;
}
.links-img a {
    margin-top: 120px;
}
/*----- Media queries */
@media screen and (max-width: 767px){
    .service-thumbnail {

```

```

        width: 100%;
        height: 200px;
    }
}

@media screen and (min-width: 767px) and (max-width: 991px){
    .service-thumbnail {
        width: 150px;
        height: 100px;
    }
}

@media screen and (max-width: 991px){
    .step-title {
        font-size: 45px;
    }
    .links-img a {
        margin-top: 48px;
    }
    .organic-img {
        height: 200px !important;
    }
    .introduction-bg {
        padding-left: 30px;
        padding-right: 0px;
    }
}

@media screen and (min-width: 768px) and (max-width: 1024px){
    .about-hero-img{
        min-height: 850px;
    }
}

```

```

    }
}
@media screen and (min-width: 1200px){
    .about-hero-img{
        min-height: 800px;
    }
}

from django.shortcuts import render, get_object_or_404, reverse, redirect
from django.contrib import messages
from profiles.models import UserProfile
from user_reviews.forms import ReviewForm
from services.models import Service
from user_reviews.models import UserReview

def about(request):
    """ A view to return about page. If a user is in session they are
    able to view the reviews and add review form. If they are not
    signed in, they are prompted to do so
    """
    service = Service.objects.all()
    reviews = UserReview.objects.all()
    review_form = ReviewForm()

    if request.user.is_authenticated:
        user = get_object_or_404(UserProfile, user=request.user)
        if request.user == user.user:
            context = {
                'user': user,
                'service': service,

```

```

        'reviews': reviews,
        'review_form': review_form,
    }
    return render(request, 'about/about.html', context)

else:
    context = {
        'service': service,
        'reviews': reviews,
    }
    return render(request, 'about/about.html', context)

def new_review(request):
    """ Allows users who have signed in to leave a review about
    a specific service that the hotel offers """
    user_profile = UserProfile.objects.get(user=request.user)
    if request.user.is_authenticated:
        if request.method == 'POST':
            review_form = ReviewForm(request.POST)
            if review_form.is_valid():
                if len(request.POST["review_content"]) <= 0 or len(
                    request.POST["service"]) <= 0:
                    messages.error(
                        request, "You haven't completed the review form! \
                            Please add content and try again.")
                return redirect(reverse("about"))
            new_review = review_form.save(commit=False)
            new_review.user_profile = user_profile

```

```

        review_form.save()
        messages.success(request, 'Success! Your review has \
                               been added.')
        return redirect(reverse("about"))
    else:
        messages.error(request, 'Your review could not be added. \
                               Please check that your review is valid.')
template = 'about/about.html'
context = {
    'review_form': review_form,
}
return render(request, template, context)

def edit_review(request, review_id):
    """ Allows the author of the review to edit what
    they have written """

    user_profile = get_object_or_404(UserProfile, user=request.user)
    review = get_object_or_404(UserReview, id=review_id)
    review_form = ReviewForm()

    if request.user == user_profile.user:
        if request.method == 'POST':
            review_form = ReviewForm(request.POST, instance=review)
            if review_form.is_valid():
                if len(request.POST["service" or "review_content"]) <= 0:
                    messages.error(
                        request, "You have not completed the review form. \

```

```

        Please add content and try again.")
    return redirect(reverse("about"))
else:
    review = review_form.save(commit=False)
    user_profile = user_profile
    review_form.save()
    messages.success(request, 'Success! Your review has \
                        been updated.')
    return redirect(reverse("about"))
else:
    review_form = ReviewForm(instance=review)
template = 'about/about.html'
context = {
    'review_form': review_form,
    'user_profile': user_profile,
    'review': review,
}
return render(request, template, context)
from django.shortcuts import render, get_object_or_404
from .models import Service

def all_stay_services(request):
    """ A view to return all Stay options to users """

    stay = Service.objects.filter(is_stay=True)
    service = Service.objects.all()

    context = {

```

```

    'stay': stay,
    'service': service,
    'stay_page': True,
}

```

```

return render(request, 'services/services.html', context)

```

```

def all_relax_services(request):

```

```

    """ A view to return all Relax options to users """

```

```

    relax = Service.objects.filter(is_relax=True)

```

```

    service = Service.objects.all()

```

```

    context = {

```

```

        'relax': relax,

```

```

        'service': service,

```

```

        'relax_page': True,

```

```

    }

```

```

    return render(request, 'services/services.html', context)

```

```

def all_eat_services(request):

```

```

    """ A view to return all Eat options to users """

```

```

    eat = Service.objects.filter(is_eat=True)

```

```

    service = Service.objects.all()

```

```

    context = {

```

```

        'eat': eat,

```

```

        'service': service,

```

```

        'eat_page': True,

```

```

    }

    return render(request, 'services/services.html', context)

def service_detail(request, service_id):
    """ A view to display individual service details, regardless
    of which category they belong to """
    service = get_object_or_404(Service, pk=service_id)

    context = {
        'service': service,
    }

    return render(request, 'services/service-detail.html', context)

from django.contrib import admin
from .models import Category, Service

class ServiceAdmin(admin.ModelAdmin):
    list_display = (
        'sku',
        'name',
        'category',
        'description',
        'image',
        'is_stay',
        'no_of_guests',
        'has_breakfast',
        'is_relax',
        'length',
        'scent',
    )

```

```

        'is_eat',
        'protein',
        'no_of_courses',
    )
    ordering = ('sku',)

```

```

class CategoryAdmin(admin.ModelAdmin):
    list_display = (
        'name',
    )

```

```

admin.site.register(Category, CategoryAdmin)
admin.site.register(Service, ServiceAdmin)
from django.db import models

```

```

class Category(models.Model):

    class Meta:
        verbose_name_plural = 'Categories'

    name = models.CharField(max_length=254)

    def __str__(self):
        return self.name

```

```

class Service(models.Model):

    sku = models.CharField(max_length=254, null=True, blank=True)

```

```
name = models.CharField(max_length=254)
category = models.ForeignKey('Category', null=True, blank=True,
                             on_delete=models.SET_NULL)
price = models.DecimalField(max_digits=6, decimal_places=2)
description = models.TextField(max_length=900)
image = models.ImageField(null=True, blank=True)

is_stay = models.BooleanField(default=False)
no_of_guests = models.IntegerField(null=True, blank=True)
has_breakfast = models.BooleanField(default=True, null=True, blank=True)

is_relax = models.BooleanField(default=False)
length = models.IntegerField(null=True, blank=True)
scents = models.CharField(max_length=254, null=True, blank=True)

is_eat = models.BooleanField(default=False)
protein = models.CharField(max_length=254, null=True, blank=True)
no_of_courses = models.IntegerField(null=True, blank=True)
def __str__(self):
    return self.name
```