

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет
імені Дмитра Моторного
Факультет енергетики і комп'ютерних технологій

ЗАТВЕРДЖУЮ

Зав. каф. "Комп'ютерні науки"

доц. _____ **Сергій ШАРОВ**
"17" лютого 2025 р.

Пояснювальна записка
до кваліфікаційної роботи здобувача СВО Магістр
(ступінь вищої освіти)

на тему: Веб застосунок для бронювання житлових та нежитлових приміщень

52/4КНД.11260394.000000ПЗ

Виконав: здобувач вищої освіти 2 курсу, групи 21МБКН
спеціальності 122 Комп'ютерні науки
за ОПП Комп'ютерні науки
(шифр і назва спеціальності та ОПП)

_____ **Володимир ТАРАСЮК**
(підпис)

Керівник, доц. _____ **Сергій ШАРОВ**
(підпис)

Нормоконтроль, ст. викл. _____ **Ольга ЗІНОВ'ЄВА**
(підпис)

Рецензент, доц. _____ **Степан СКРУПСЬКИЙ**
(підпис)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет імені Дмитра
Моторного

Факультет: Енергетики і комп'ютерних технологій

Кафедра: Комп'ютерні науки

Ступінь вищої освіти: Магістр

Спеціальність: 122 Комп'ютерні науки

ОПП: Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

к.пед.н., доц. Сергій ШАРОВ

“04” жовтня 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) ЗДОБУВАЧУ ВО

Тарасюку Володимиру Романовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Веб застосунок для бронювання житлових та нежитлових приміщень» _____

керівник проєкту Шаров Сергій Володимирович, к.пед.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом університету від “23” вересня 2024 року № 442-С

2. Строк подання здобувачем вищої освіти роботи 14 лютого 2025

3. Вихідні дані до роботи: дані обстеження об'єкту, статистичні дані, технічне завдання на дипломне проєктування, матеріали виробничих практик, нормативні документи, науково-технічна література, електронні ресурси та ін.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

1. Аналіз предметної області та існуючих рішень.

2. Специфікація вимог до веб застосунку

3. Обґрунтування мови та середовища програмування.

4. Дослідна експлуатація веб застосунку для бронювання житлових та нежитлових приміщень

5. Тестування веб застосунку для бронювання житлових та нежитлових приміщень

5. Перелік графічного матеріалу

«Слайд 1. Титульний слайд (тема, ПІБ здобувача освіти, ПІБ наукового керівника)»,

«Слайд 2. Предмет, об'єкт, мета, завдання дослідження»,

«Слайд 3. Порівняльна характеристика аналогів програмного продукту »,
 «Слайд 4. Діаграма варіантів використання»,
 «Слайд 5. Інструментальні засоби (мова програмування, середовище програмування, база даних)»,
 «Слайд 6. Інтерфейс і робота програмного продукту»,
 «Слайд 7. Інтерфейс і робота програмного продукту»,
 «Слайд 8. Тестування»,
 «Слайд 9. Висновки»,
 «Слайд 10. Дякую за увагу»

6. Консультанти розділів проєкту (роботи)

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

7. Дата видачі завдання 01 жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів роботи	Примітка
	Аналіз предметної області	12.09.2024	Виконано
	Специфікація вимог до <u>веб застосунку</u>	06.10.2024	Виконано
	Проектування веб застосунку	16.10.2024	Виконано
	Обґрунтування інструментальних засобів	05.11.2024	Виконано
	Розробка веб застосунку для бронювання житлових та нежитлових приміщень	10.12.2024	Виконано
	Тестування та налагодження <u>веб застосунку</u>	20.01.2025	Виконано
	Підпис керівником роботи	14.02.2025	Виконано
	Підпис завідувачем кафедри	17.02.2025	Виконано

Здобувач вищої освіти _____ Володимир ТАРАСЮК _____
 (підпис) (власне ім'я та ПРІЗВИЩЕ)

**Керівник
кваліфікаційної
роботи**

_____ Сергій ШАРОВ _____
 (підпис) (власне ім'я та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра: 92 с., 9 рис., 6 табл., 43 посилань, 3 додатки.

Актуальність. У сучасному світі цифрових технологій зростає попит на зручні та доступні платформи для здійснення різноманітних операцій в режимі онлайн. Однією з важливих галузей є оренда приміщень, як комерційних, так і житлових. Ринок оренди швидко трансформується завдяки впровадженню інформаційних технологій, що дозволяє забезпечити оперативний пошук, порівняння та бронювання приміщень. Створення веб-застосунку для оренди приміщень сприятиме підвищенню ефективності управління орендними процесами, зменшенню часу на пошук потрібного приміщення та підвищенню рівня задоволеності як орендарів, так і власників нерухомості.

Об'єкт дослідження: процес електронного пошуку та бронювання житлових та нежитлових приміщень.

Предмет дослідження: веб застосунок для бронювання житлових та нежитлових приміщень.

Мета роботи: Розробка веб-застосунку для оренди приміщень, який дозволить оптимізувати процес пошуку, вибору та бронювання приміщень, забезпечуючи зручний інтерфейс та високу якість обслуговування користувачів.

Задачі дослідження:

1. Провести аналіз сучасних рішень на ринку веб-платформ для оренди приміщень та визначити їх переваги і недоліки.
2. Розглянути технологічні інструменти та фреймворки для розробки веб-застосунків, зокрема Next.js, React, Supabase та інші.
3. Розробити технічне завдання, що містить вимоги до функціональних можливостей, безпеки та продуктивності застосунку.

4. Створити концептуальну модель системи, визначити архітектуру та розробити алгоритми взаємодії між користувачами та серверною частиною.
5. Виконати програмну реалізацію веб-застосунку, скласти детальну інструкцію користувача та провести тестування системи.

Практичне значення роботи полягає у впровадженні розробленого веб-застосунку, що дозволить оптимізувати процес оренди приміщень. Завдяки інтегрованій системі онлайн бронювання, платформа сприятиме прозорості інформації, зменшенню витрат часу на пошук та укладання договорів, а також покращенню комунікації між орендодавцями та орендарями. Реалізація запропонованого рішення сприятиме підвищенню конкурентоспроможності на ринку нерухомості та створенню додаткової вартості для учасників ринку.

Ключові слова: ВЕБ-ЗАСТОСУНОК, ОRENDA ПРИМІЩЕНЬ, БРОНЮВАННЯ, ЕЛЕКТРОННА КОМЕРЦІЯ, ІНТЕРАКТИВНИЙ ІНТЕРФЕЙС.

SUMMARY

Master's thesis: 92 p., 9 figures, 6 tables, 43 references, 3 appendices.

Relevance. In today's world of digital technologies, there is a growing demand for convenient and affordable platforms for conducting various online transactions. One of the most important industries is the rental of premises, both commercial and residential. The rental market is rapidly transforming due to the introduction of information technology, which allows for the prompt search, comparison and booking of premises. The creation of a web application for renting premises will help to increase the efficiency of managing rental processes, reduce the time spent searching for the right premises and increase the level of satisfaction of both tenants and property owners.

Research object: the process of electronic search and booking of residential and non-residential premises.

Subject of research: a web application for booking residential and non-residential premises.

Purpose of the study: Development of a web application for renting premises that will optimize the process of searching, selecting and booking premises, providing a user-friendly interface and high quality of user service.

Research objectives:

1. To analyze current solutions in the market of web-based platforms for renting premises and identify their advantages and disadvantages.
- 2) To consider technological tools and frameworks for developing web applications, including Next.js, React, Supabase, and others.
- 3) Develop a technical task that contains requirements for the functionality, security, and performance of the application.
- 4) Create a conceptual model of the system, define the architecture and develop algorithms for interaction between users and the server side.
- 5) Perform software implementation of the web application, create a detailed user manual and test the system.

The practical significance of the work lies in the implementation of the developed web application, which will optimize the process of renting premises. Thanks to the integrated online booking system, the platform will promote transparency of information, reduce the time spent on searching and concluding contracts, and improve communication between landlords and tenants. The implementation of the proposed solution will help to increase competitiveness in the real estate market and create additional value for market participants.

Keywords: WEB APPLICATION, RENTAL OF PREMISES, BOOKING, E-COMMERCE, INTERACTIVE INTERFACE.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	13
1.1 Використання ІКТ у готельному бізнесі та ринку нерухомості.....	13
1.2 Аналіз існуючих рішень для бронювання житлової та нежитлової нерухомості.....	18
1.3. Технічне завдання на розробку веб-застосунку для бронювання житлових та нежитлових приміщень	31
РОЗДІЛ 2 ПРОЕКТУВАННЯ Веб-застосунку для бронювання житлових приміщень	34
2.1 Концептуальна модель використання веб-застосунку для бронювання житлових приміщень.	34
2.2 Функціональна модель веб-застосунку для бронювання житлових приміщень.	41
2.3. Специфікація функціональних та нефункціональних вимог	43
РОЗДІЛ 3 ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ Веб- застосунку для бронювання житлових приміщень.....	48
3.1 Загальний огляд технологічного стеку	48
3.2 Обґрунтування вибору мови програмування	52
3.3 Інструменти для розробки та CI/CD.....	58
РОЗДІЛ 4 ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ	61
4.1 Особливості програмної реалізації.....	61
4.2 Інструкція користувача веб-застосунку для оренди приміщень.....	66
4.3 Тестування веб-застосунку	69
4.4. Розгортання веб-застосунку.....	73
ВИСНОВКИ.....	76
СПИСОК ЛІТЕРАТУРИ.....	78
ДОДАТКИ.....	83

ВСТУП

Актуальність. На сьогодні інформаційно-комунікаційні технології (ІКТ) стрімко розвиваються, охоплюючи майже всі сфери сучасного життя. Однією з важливих галузей є нерухомість, де процес оренди приміщень набуває нових вимірів завдяки застосуванню сучасних цифрових рішень. Веб-застосунки для оренди приміщень дозволяють автоматизувати пошук, бронювання та управління нерухомістю, знижуючи витрати часу та ресурсів як для орендодавців, так і для орендарів. Застосування таких систем сприяє підвищенню прозорості ринку, оптимізації комунікації між учасниками процесу та забезпеченню високої якості обслуговування.

У зв'язку з актуальністю цифровізації процесів в сфері нерухомості, тему дослідження було сформульовано як «Розробка веб-застосунку для оренди приміщень».

Об'єкт дослідження: процес електронного пошуку та бронювання житлових та нежитлових приміщень.

Предмет дослідження: веб застосунок для бронювання житлових та нежитлових приміщень.

Мета роботи: Розробка веб-застосунку для оренди приміщень, який дозволить оптимізувати процес пошуку, вибору та бронювання приміщень, забезпечуючи зручний інтерфейс та високу якість обслуговування користувачів.

Сформована мета, об'єкт і предмет дослідження дозволили поставити такі завдання:

1. Провести аналіз сучасних рішень на ринку веб-платформ для оренди приміщень та визначити їх переваги і недоліки.
2. Розглянути технологічні інструменти та фреймворки для розробки веб-застосунків, зокрема Next.js, React, Supabase та інші.
3. Розробити технічне завдання, що містить вимоги до функціональних можливостей, безпеки та продуктивності застосунку.

4. Створити концептуальну модель системи, визначити архітектуру та розробити алгоритми взаємодії між користувачами та серверною частиною.
5. Виконати програмну реалізацію веб-застосунку, скласти детальну інструкцію користувача та провести тестування системи.

При реалізації поставлених задач було використано наступні методи дослідження: теоретичні – аналіз наукової та спеціальної літератури з питань розробки веб-застосунків, сучасних технологій та тенденцій на ринку нерухомості; практичні – розробка програмного забезпечення з використанням сучасних технологій, моделювання архітектури системи, тестування функціональних можливостей застосунку та аналіз отриманих результатів.

Практичне значення дослідження полягає у створенні веб-застосунку, який дозволить автоматизувати процес оренди приміщень, забезпечити зручну комунікацію між орендодавцями та орендарями, а також сприятиме підвищенню ефективності управління нерухомістю. Результати роботи можуть бути впроваджені як у великих компаніях, так і серед малих підприємств, що займаються орендою нерухомості, сприяючи розвитку цифрової економіки в галузі нерухомості.

Апробація результатів здійснювалася шляхом представлення доповіді на Всеукраїнській науково-практичній конференції з інформаційних технологій та публікації статті «Сучасні веб-технології в автоматизації процесів оренди приміщень» у спеціалізованому виданні.

Структура і обсяг роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел (43 джерела) та 3 додатків. Перший розділ присвячено аналізу сучасного ринку оренди приміщень та дослідженню існуючих веб-платформ, що забезпечують цей процес. У ньому також розглянуто проблематику традиційних методів пошуку та бронювання нерухомості. Другий розділ містить розробку технічного завдання, формулювання функціональних і нефункціональних

вимог до системи, а також опис концептуальної моделі застосунку. У третьому розділі детально розглянуто технологічний стек, обґрунтовано вибір мови програмування та інструментів розробки, а також описано архітектуру системи. Четвертий розділ присвячено реалізації веб-застосунку, його тестуванню, складанню інструкції користувача та аналізу отриманих результатів. Загальний обсяг роботи становить 92 сторінки.

РОЗДІЛ 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Використання ІКТ у готельному бізнесі та ринку нерухомості

В умовах інформатизації суспільства та впровадження інформаційно-комунікаційних технологій (ІКТ) у різні напрямки діяльності та виробництва широке розповсюдження отримали різноманітні онлайн сервіси. Їх використання дозволяє залучити більшу цільову аудиторію, покращити маркетингову діяльність, здійснювати онлайн комерцію тощо. Важливими напрямками застосування ІКТ є підтримка діяльності туристичного бізнесу, а також ринку нерухомості. В межах цих напрямків клієнти можуть здійснювати бронювання та аренду житлових та нежитлових приміщень, відслідковувати статус замовлень, комунікувати з орендодавцями та туристичними операторами тощо.

В умовах жорсткої конкуренції підприємства запроваджують нові форми взаємодії з клієнтами, сучасні засоби реалізації товарів та послуг через мережу Інтернет. І це є об'єктивним кроком на шляху підвищення власної конкурентоспроможності, адже інтернет-ресурсів зростає з кожним днем. Добре продуманий та просунутий веб ресурс дозволяє отримати більший прибуток, ніж робота декількох операторів (співробітників підприємства), які самотужки шукають клієнтів та партнерів, здійснюють комунікацію тощо. Крім того, у разі використання інтернет-ресурсів відсутні витрати на оренду дороги, проживання під час відрядження, інших витрат [31, с. 105].

Важливими напрямками застосування ІКТ у всьому їх різноманітті є ринок нерухомості та туристичний бізнес, які в сучасних умовах знаходяться в динамічному розвитку, не завжди позитивному. Ці два напрямки мають принаймні одну точку перетину – оренда або бронювання житлової нерухомості. І від того, наскільки якісно буде забезпечено інформування

клієнтів про житло, умови бронювання/оренду, способи оплати тощо, буде напряду залежати успішність оформлених угод.

Одночасне функціонування ринку нерухомості, ринку товарів та послуг, ринку капіталів визначає особливості господарського механізму регульованої економіки. Як наслідок, ці три складові є одними із ключових сегментів сучасної економіки. Розвиток ринку нерухомості відбувається в умовах перехідної економіки та характеризується нерівномірністю, недосконалістю законодавчої бази, низькою платоспроможністю населення [15, с. 67], нерівномірністю сегментації ринку, проблеми адаптації до сучасних економічних умов [17, с. 147-154] та ін. Тобто спостерігаються певні труднощі у функціонуванні ринку нерухомості, що відтворюється у періодичні спадах і піднесеннях.

Якщо розглянути можливості ІКТ для оптимізації функціонування ринку нерухомості та підвищення його ефективності, то можна виокремити декілька переваг. Використання онлайн сервісів та мобільних додатків для висвітлення в мережі Інтернет оголошень про оренду житла. У випадку житлових приміщень потенційні клієнти орендарі можуть переглянути деталі житла, здійснити фільтрацію знайдених варіантів, прочитати відгуки тощо. Якщо потрібно знайти нежитлове приміщення (наприклад, офісне або складське приміщення), то за допомогою параметрів пошуку можна швидко знайти відповідні приміщення відповідно до потреб компаній.

Оскільки для онлайн сервісів практично немає обмежень у поширенні інформації про житловий та нежитловий фонд серед цільової аудиторії, це дозволяє орендодавцям вийти на регіональний або навіть міжнародний ринок. Потенційні клієнти можуть орендувати житло або офіси в інших країнах, використовуючи міжнародні платформи Airbnb, Booking.com, Zillow та ін. Це надає можливість збільшити діапазон житла та підібрати квартиру за оптимальною ціною. У роботі [24, с. 118] звертається увага на таких платформах як ЛУН, Dom Ria, Zillow.com, які широко використовуються в українському сегменті.

Онлайн платформа ЛУН орієнтована виключно на житлову нерухомість, позиціонує себе як найбільший український сайт для пошуку квартир. За своїми функційними особливостями він «агрегатором», тобто збирає інформацію з аналогічних українських платформ та відображає її у зручному форматі. Основною перевагою ЛУН є автоматична генерація даних шляхом збору інформації з інших джерел. Це дозволяє не створювати окремі форми для введення даних про житло. Крім того, сервіс пропонує широкий набір фільтрів, що значно спрощує пошук потрібного об'єкта.

Сервіс Dom Ria зосереджений на житловій нерухомості, накопичує інформацію після її введення безпосередньо користувачами. Переваги даного сервісу полягають в тому, що можна викликати інспектора з нерухомості, який допомагає оцінити житло та визначити ринкову вартість. До недоліків слід віднести обмежений набір фільтрів, що може ускладнювати пошук потрібного об'єкта.

Провідний міжнародний сервіс Zillow призначений для пошуку житлової нерухомості. Zillow охоплює всі аспекти життєвого циклу володіння нерухомістю, включаючи купівлю, продаж, оренду, фінансування та реконструкцію, що робить його комплексним інструментом для управління житлом. Zillow дозволяє отримувати не лише інформацію про житло, але й забезпечує комунікацію з місцевими професіоналами, які нададуть консультації щодо оздоблення квартири, садівництва в домі чи ремонту житла.

Через використання спеціалізованих онлайн платформ можна досягти певного рівня прозорості та довіри до орендодавців. Такі сервіси часто включають рейтингові системи серед орендодавців і орендарів, що дозволяє уникати недоброчесних угод. Крім того, зазвичай такі платформи містять достатньо детальну інформацію про житлові та нежитлові приміщення, зокрема дані про попередніх орендарів, технічний стан приміщення, юридичні аспекти угоди тощо. Прикладом може слугувати система електронних закупівель "Prozorro", що була розроблена з метою підвищення прозорості та

підзвітності державних закупівель. Онлайн платформа дозволяє здійснювати всі етапи закупівельного процесу в електронному вигляді, забезпечуючи рівні права для всіх учасників торгів. Для фізичних осіб широким попитом користується платформа “Дія”, що дозволяє отримувати значну кількість адміністративних послуг, не виходячи з дому та не витрачаючи часу [11, 892].

Наступним напрямком економічної діяльності, де можна застосувати інформаційно-комунікаційні технології для оренди житлової нерухомості, є туристичний бізнес. Науковці зазначають, що готельний та ресторанний бізнес є однією з галузей, яка найбільш динамічно розвивається в напрямі застосування ІКТ. Актуальним є впровадження інновацій для створення нових конкурентоспроможних туристичних продуктів, формування сприятливих умов для залучення інвестицій та активізації туристичної діяльності [12, с. 456]. Як наслідок, інформаційні технології у сфері туризму дозволяють користувачу самостійно планувати подорож, спілкуватися з місцевими жителями за допомогою синхронного перекладу через мобільний перекладач, отримувати персоналізовані рекомендації щодо відвідування туристичних об’єктів та ін. Крім того, впровадження ІКТ у туристичну індустрію є закономірним кроком на шляху цифровізації туристичної діяльності, що відповідає світовим стандартам розвитку передових технологій та оновлення туристичних послуг [1, с. 21].

Туристичні компанії для підвищення конкурентоспроможності серед анархічних туроператорів активно застосовують інтернет-технології для досягнення двох основних цілей: маркетингу та прямих продажів своїх послуг [18, с. 386]. З метою підвищення ефективності обслуговування клієнтів і розширення асортименту послуг вони використовують спеціалізоване програмне забезпечення, у тому числі засноване на використанні штучного інтелекту [21, с. 70]. Наприклад, окремі програмні продукти здатні формувати індивідуальні тури, що включають проживання, страхування, трансфер, екскурсії та інші опції, які туристи можуть одразу

забронювати під час замовлення туру. За допомогою спеціалізованого прикладного програмного забезпечення можна швидко знайти оптимальний варіант відпочинку за різними критеріями відбору, контролювати заявку з боку туроператора, зменшити кількість об'єктивних та суб'єктивних помилок, зменшити вартість туру за умова завчасного замовлення. До недоліків можна віднести високу вартість програмного продукту та обмежена доступність для туристичних компаній порівняно з глобальними чи альтернативними системами бронювання [19].

З іншого боку, можна скористатися значними перевагами, що надає мережа Інтернет. Хоча традиційні джерела інформації, такі як туристичні агенти, телебачення та журнали, залишаються актуальними, все більше туристів віддають перевагу онлайн-замовленням послуг, використанню онлайн-туристичних агентств та самостійному бронюванню в авіакомпаніях, готелях і компаніях із прокату автомобілів [1, с. 20]. Перспективи розвитку індустрії гостинності тісно пов'язані із зростанням ролі онлайн бронювань та соціальних мереж, які стають основними каналами для резервування готелів і ресторанів [9, с. 5-7]. Одним із відомих онлайн сервісів з бронювання житла є Booking.com. Перевага сайту полягає у можливості не оплачувати проживання заздалегідь, а здійснювати оплату безпосередньо під час заселення. Крім того, сервіс має гнучку систему знижок на бронювання житла. Такий підхід у бронювання житла забезпечив моделі Booking.com домінуючі позиції на ринку бронювання та оренди нерухомості [19].

Останнім часом в туристичному бізнесі стали застосовуватися віртуальні та доповнені технології. За допомогою 3D-турів можна ознайомитися з туристичним об'єктом, житловим приміщенням тощо. Це особливо корисно у випадку міжнародних поїздок, де немає можливості особисто подивитися об'єкт перед бронюванням. А використання технологій доповненої реальності дозволяють побачити, як виглядає приміщення після ремонту або меблювання.

Отже, впровадження ІКТ у ринок нерухомості дозволяє зробити його більш зручним, прозорим і доступним як для орендодавців, так і для орендарів. Інформаційно-комунікаційні технології відкривають нові перспективи для розвитку, полегшують доступ до об'єктів нерухомості та підвищують ефективність взаємодії між сторонами.

Впровадження та використання ІКТ у готельному бізнесі надає можливість підвищити рівень інформування, обслуговування та задоволеності клієнтів. Завдяки персоналізації послуг, автоматизації окремих операцій та аналізу даних заклади готельно-ресторанного бізнесу отримують нові можливості для залучення нових та утримування постійних клієнтів, таким чином забезпечуючи високу конкуренцію серед аналогічних компаній.

1.2 Аналіз існуючих рішень для бронювання житлової та нежитлової нерухомості

У процесі аналізу предметної області було визначено, що актуальним є створення веб-застосунку, який дозволить користувачам знаходити об'єкти нерухомості за заданими критеріями. Планується розробити такий застосунок, а також, за можливості, інтегрувати його з популярними сервісами, щоб аналізувати їхні пропозиції. Якщо ці пропозиції відповідатимуть запитам клієнта, вони будуть автоматично надсилатися йому.

Для цього потрібно розібратися, що таке нерухомість, які її види існують та чому вона має таке значення.

Нерухомість – це майно фізичної або юридичної особи, яке неможливо перемістити без пошкоджень. Все інше майно вважається рухомим. До нерухомості належать житлові та нежитлові приміщення, споруди, земельні ділянки тощо. Якщо об'єкт нерухомості закріплений фундаментом на земельній ділянці, він називається капітальним. У протилежному випадку це некапітальна нерухомість.

Об'єкти нерухомості можуть використовуватися для проживання та реєстрації за адресою, а також для отримання пасивного доходу від бізнесу. Залежно від призначення, нерухомість поділяється на житлову та нежитлову, а також має інші категорії. Земельні ділянки є основою для всіх видів нерухомості. Їх часто купують забудовники, об'єднують з іншими об'єктами та переплановують, щоб збільшити щільність забудови та вартість. [2]

Житлова нерухомість включає житло для окремих осіб, сімей або груп людей. Це найпоширеніший тип нерухомості, знайомий багатьом. Прикладами є односімейні будинки, квартири, таунхауси тощо. Комерційна нерухомість охоплює землі та будівлі, які використовуються для бізнесу, наприклад, торгові центри, офіси, готелі, автостоянки та медичні заклади.

Промислова нерухомість стосується об'єктів, які використовуються для виробничих, дослідницьких, логістичних, складських та будівельних цілей.

Ця інформація необхідна для чіткого розуміння, з якими даними буде працювати розроблюваний веб-застосунок. Очікується, що платформа забезпечить усі необхідні функції для швидкого пошуку нерухомості та задовольнить потреби користувачів [6; 7].

Booking.com – це одна з найпопулярніших онлайн-платформ для пошуку, бронювання та оренди житла по всьому світу. Компанія була заснована в 1996 році та наразі є частиною Booking Holdings, що є лідером у сфері туристичних послуг (рис. 1.1).

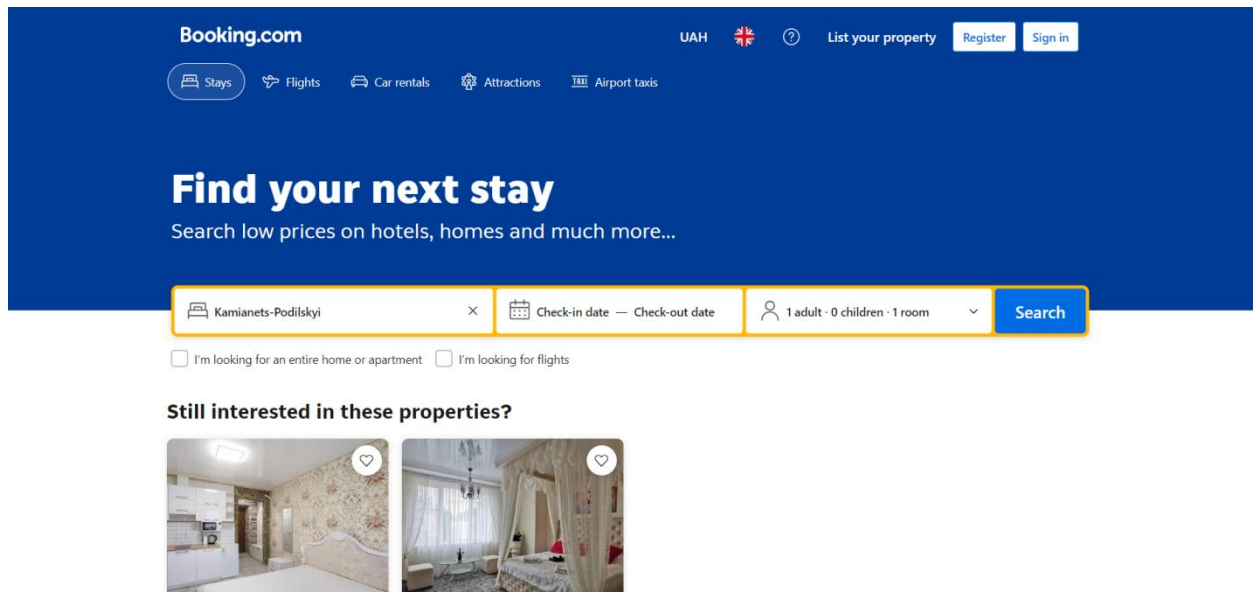


Рисунок 1.1 – Інтерфейс сайту Booking.com

Платформа надає доступ до мільйонів пропозицій від готелів, апартаментів, хостелів, будинків для відпочинку, вілл, кемпінгів та інших типів житла [29]. Основні функції платформи:

- пошук житла. Booking.com дозволяє користувачам шукати житло за різними критеріями: розташування (місто, країна, регіон або конкретна адреса); тип житла (готель, квартира, хостел тощо); ціновий діапазон; кількість гостей та номерів; додаткові зручності (Wi-Fi, сніданок, басейн, паркінг тощо);

- фільтри для уточнення пошуку. Booking.com дозволяє користувачам обирати з широкого набору фільтрів, щоб швидко знайти житло, що відповідає їхнім уподобанням;

- інтерактивна карта. Booking.com дає змогу переглядати варіанти житла на інтерактивній мапі, що дозволяє оцінити зручність розташування та близькість до цікавих місць і транспортних вузлів;

- огляди та рейтинги. Booking.com використовує систему відгуків і рейтингів від реальних користувачів, що допомагає оцінити якість послуг, чистоту, комфорт та інші важливі аспекти;

- програма лояльності Genius. Booking.com пропонує знижки та спеціальні пропозиції для постійних клієнтів у рамках програми лояльності Genius;

- інтеграція з іншими послугами. Booking.com дозволяє бронювати перельоти, орендувати автомобілі, замовляти екскурсії та інші туристичні послуги, створюючи комплексне рішення для організації подорожей;

Особливості бізнес-моделі:

- платформа діє як посередник між постачальниками послуг (готелями, орендодавцями) та користувачами;

- Booking.com отримує комісію за кожне бронювання, що здійснюється через платформу;

- великий акцент робиться на зручності користувацького інтерфейсу, локалізації для різних ринків та багатомовній підтримці.

RoomGuru.com – це метапошуковий сервіс, який допомагає користувачам знайти найкращі пропозиції на бронювання житла, порівнюючи ціни з різних онлайн-платформ [36]. Його особливість полягає в можливості швидкого аналізу вартості житла з популярних сайтів бронювання, таких як Booking.com, Agoda, Expedia, Hotels.com та інших (рис. 1.2).

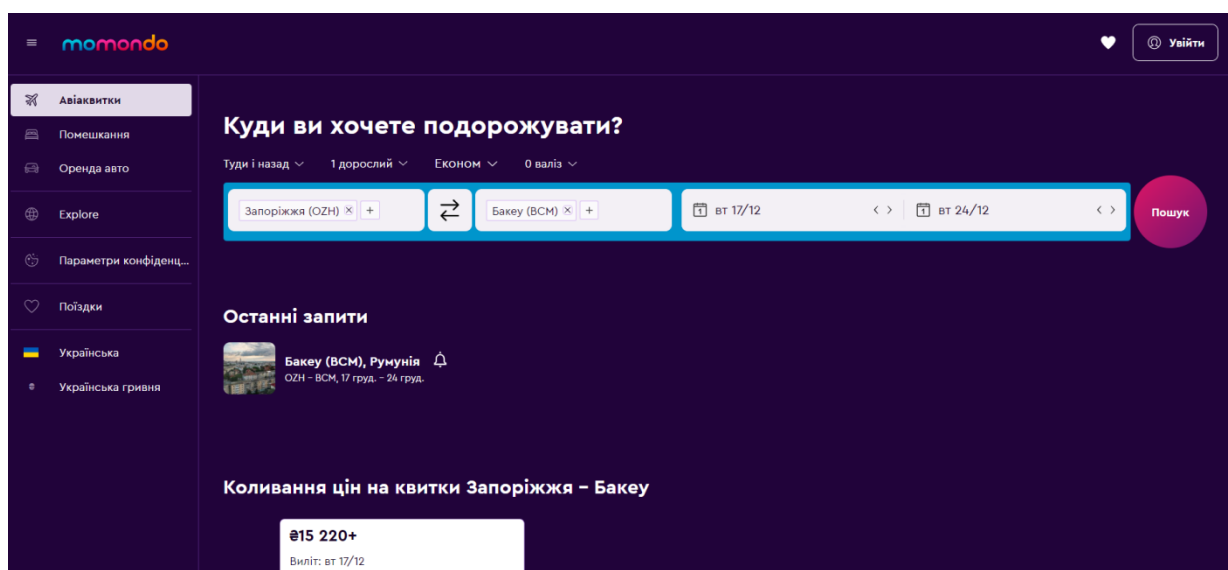


Рисунок 1.2 – Інтерфейс сайту “RoomGuru”

RoomGuru.com спрямований на спрощення процесу пошуку житла, пропонуючи користувачам можливість знаходити найбільш вигідні пропозиції. Це досягається завдяки збору даних з різних джерел і наданню користувачам зручного інструменту для порівняння цін.

Функціональні можливості:

- RoomGuru.com дозволяє користувачам шукати готелі, апартаменти, хостели, будинки для відпочинку та інші види житла. Пошук можливий за ключовими параметрами, такими як: місцезнаходження (місто, країна, район); дати перебування; кількість гостей;

- порівняння цін. Основна функція RoomGuru – це відображення цін на одні й ті самі пропозиції житла з різних платформ. Це дає змогу користувачам знайти найдешевший варіант і уникнути переоплат;

- інтерактивна карта. Інтегрована карта допомагає користувачам орієнтуватися в розташуванні об'єктів нерухомості та вибирати зручні варіанти відповідно до місця перебування або віддаленості від визначних місць;

- фільтри та сортування. RoomGuru надає широкий набір фільтрів, що дозволяє уточнювати пошук за категоріями, ціною, рейтингом, наявністю зручностей та іншими параметрами;

- рейтинги та відгуки. Інформація про рейтинги та відгуки з різних платформ допомагає користувачам приймати обґрунтовані рішення щодо бронювання.

Особливості бізнес-моделі:

- RoomGuru не здійснює прямого бронювання. Замість цього сервіс перенаправляє користувачів на платформи-партнери для оформлення замовлення.;

- компанія отримує дохід за рахунок партнерських програм з платформ, які платять за залучення клієнтів через RoomGuru;

- завдяки співпраці з великою кількістю платформ, сервіс забезпечує доступ до пропозицій у більш ніж 200 країнах.

RoomGuru.com має декілька переваг. По-перше, користувачі отримують змогу швидко знайти найвигідніші ціни без необхідності відвідувати кожну платформу окремо. По-друге, інтуїтивно зрозумілий інтерфейс спрощує процес пошуку. По-третє, порівняння здійснюється на основі актуальних даних, що знижує ризик прихованих витрат.

Обмеження RoomGuru полягає в тому, що платформа не відповідає за якість послуг, наданих сторонніми платформами, оскільки він виступає лише як посередник. Деякі унікальні пропозиції можуть бути недоступними через обмеження партнерської бази.

Hotels.com – це одна з провідних онлайн-платформ для пошуку, порівняння та бронювання готелів, апартаментів, курортів та інших видів житла по всьому світу [37]. Компанія є частиною групи Expedia Group і працює в понад 200 країнах. Вона надає користувачам доступ до великої бази даних житла, що включає різноманітні варіанти для подорожей як бізнес-цілей, так і відпочинку (рис. 1.3).

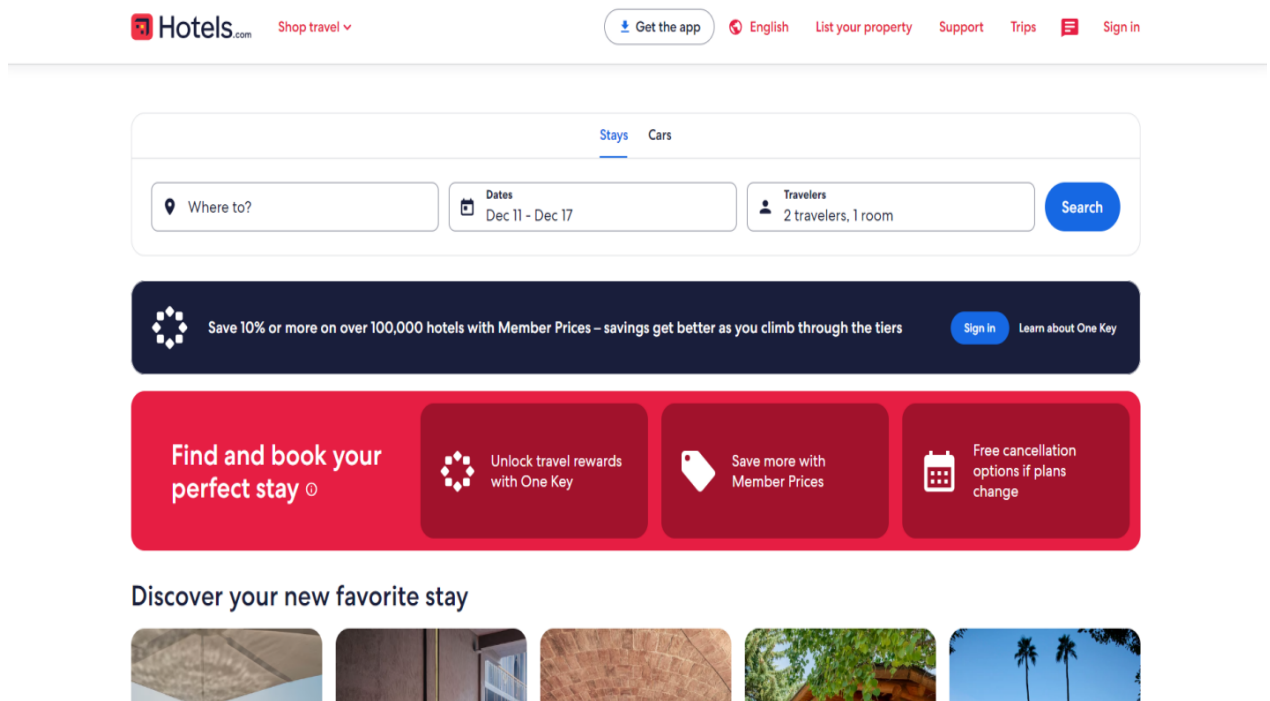


Рисунок 1.3 – інтерфейс сайту “Hotels”

Hotels.com допомагає спростити процес бронювання житла, надаючи широкий вибір пропозицій, зручний інтерфейс і програму лояльності для заохочення постійних клієнтів. Hotels.com дозволяє шукати житло за такими критеріями: місце розташування (країна, місто, район або навіть орієнтир); дати перебування; кількість гостей; тип житла (готелі, апартаменти, курорти тощо).

Фільтри дозволяють уточнювати результати пошуку за ціною, кількістю зірок, оцінками гостей, наявністю зручностей (Wi-Fi, басейн, кухня) та іншими параметрами. Користувачі можуть переглядати варіанти житла на інтерактивній мапі, що спрощує вибір з урахуванням місцезнаходження. На платформі доступні відгуки реальних гостей та рейтинги, що допомагає оцінити якість житла та послуг. Hotels.com регулярно пропонує знижки, акції та пропозиції «секретних цін» для зареєстрованих користувачів.

Програма лояльності – Hotels.com Rewards полягає у наступному: за кожні 10 ночей, заброньованих через платформу, користувач отримує 1 безкоштовну ніч (вартість залежить від середньої ціни заброньованих ночей). Ця програма заохочує постійних клієнтів і робить сервіс вигідним для частих мандрівників.

Особливості бізнес-моделі:

- Hotels.com виступає посередником між постачальниками житла (готелями, орендодавцями) та користувачами;
- компанія отримує комісію за кожне бронювання, здійснене через платформу;
- готелі можуть просувати свої пропозиції через рекламу на платформі.
- стимулює повернення клієнтів, збільшуючи обсяг повторних бронювань.

Переваги сервісу полягають у наступному:

- доступ до мільйонів об'єктів у різних країнах;
- інтуїтивно зрозумілий дизайн для простого користування;

- мотивує до використання платформи для регулярного бронювання;
- доступ до скасування бронювання, вибору з додаткових послуг тощо.

Недоліки:

- безкоштовні ночі покривають лише середню вартість попередніх бронювань, що може не підійти для дорожчих об'єктів;
- ціни можуть змінюватися залежно від партнерських умов.

Hotels.com є зразком успішної платформи для бронювання житла, яка пропонує комплексний сервіс для туристів. Її модель бізнесу та акцент на лояльність клієнтів роблять її конкурентоспроможною та популярною серед мандрівників.

Lun.ua – це онлайн-платформа для пошуку нерухомості в Україні. Вона орієнтована на користувачів, які шукають варіанти покупки, оренди або інвестування в житло [38]. Lun.ua спеціалізується на агрегації пропозицій з різних джерел, надаючи користувачам зручний інструмент для швидкого і ефективного пошуку.

Платформа допомагає спростити процес вибору житла завдяки агрегуванню даних з великої кількості ресурсів (зокрема, з сайтів забудовників і агентств нерухомості) та зручному інтерфейсу для пошуку й аналізу нерухомості.

Lun.ua пропонує користувачам шукати об'єкти за такими критеріями:

- тип нерухомості (первинний або вторинний ринок);
- розташування (місто, район, конкретна адреса);
- тип житла (квартири, будинки, комерційна нерухомість).;
- ціна (загальна вартість або вартість за квадратний метр);
- площа та кількість кімнат.

Платформа пропонує перегляд об'єктів нерухомості на карті, що дозволяє оцінити зручність розташування об'єктів, доступність інфраструктури, транспортну розв'язку та віддаленість від ключових місць.

Lun.ua збирає дані з різних ресурсів, таких як сайти забудовників,

агентства нерухомості та приватні оголошення. Це значно скорочує час, необхідний для аналізу ринку. Платформа відслідковує оновлення інформації про об'єкти, видаляючи неактуальні оголошення.

Lun.ua надає користувачам доступ до даних про динаміку цін, популярні райони, типові характеристики житла тощо. Платформа приділяє особливу увагу новобудовам, надаючи детальну інформацію про забудовників, умови придбання, інфраструктуру та хід будівництва.

Особливості бізнес-моделі заключаються в тому, що Lun.ua не продає нерухомість безпосередньо, а виступає посередником, що надає інформацію. Монетизація реалізується через продаж реклами забудовникам та агентствам нерухомості та просування преміальних оголошень на платформі. Крім того, основні функції платформи доступні користувачам безкоштовно, що збільшує залучення аудиторії.

Переваги Lun.ua:

- швидкий доступ до великої кількості пропозицій на одному ресурсі;
- допомагає користувачам оцінити розташування та оточення нерухомості;
- простота у використанні як для професіоналів, так і для звичайних користувачів;
- інформація про ринок допомагає користувачам приймати обґрунтовані рішення;
- забезпечує детальну інформацію для інвесторів і покупців.

Недоліки:

- дані платформи залежать від актуальності інформації, яку надають партнери;
- у деяких містах доступність об'єктів може бути меншою через специфіку ринку;
- Lun.ua виступає лише інформаційним посередником і не забезпечує можливості укладення угоди безпосередньо через платформу.

Lun.ua відіграє важливу роль у спрощенні пошуку та вибору нерухомості, об'єднуючи ключових гравців ринку – забудовників, агентства

нерухомості та покупців. Платформа підвищує прозорість ринку завдяки деталізації даних та зручному доступу до інформації.

Zillow.com – це провідний ринок нерухомості в Сполучених Штатах. Він пропонує комплексну платформу для купівлі, продажу, оренди та фінансування житла [43]. Zillow надає користувачам широкий спектр інструментів та ресурсів для навігації на ринку нерухомості, включаючи:

- Zillow містить мільйони оголошень про продаж та оренду, пропонуючи детальну інформацію про кожен об'єкт, включаючи фотографії, описи та ціни;
- Zillow має власний алгоритм Zestimate, який оцінює ринкову вартість будинку на основі різних даних;
- пропонує захоплюючі 3D-тури по будинках та інтерактивні плани поверхів, щоб користувачі могли краще ознайомитися з об'єктами, які їх цікавлять;
- платформа надає варіанти іпотеки та з'єднує користувачів з кредитними офіцерами, щоб допомогти їм фінансувати свої будинки;
- Zillow з'єднує користувачів з місцевими агентами з нерухомості, які можуть допомогти їм з купівлею або продажем будинку.

Zillow прагне спростити процес роботи з нерухомістю, пропонуючи безперервний досвід для орендарів, покупців та продавців. Ви можете дізнатися більше про Zillow на їхньому офіційному веб-сайті.

Dom.gia – це провідний український онлайн-ресурс для купівлі, продажу та оренди нерухомості [39]. Платформа пропонує широкий спектр послуг та інструментів для користувачів, які шукають житлову або комерційну нерухомість. Ось основні можливості Dom.gia:

- Dom.gia містить тисячі оголошень про продаж та оренду квартир, будинків, земельних ділянок та комерційної нерухомості;
- всі оголошення на сайті перевіряються інспекторами, які роблять панорамні фото 360°, перевіряють адресу, метраж та планування квартир, щоб забезпечити безпеку та достовірність інформації;

- платформа пропонує зручні фільтри для швидкого пошуку нерухомості за різними параметрами, такими як ціна, розташування, тип нерухомості тощо;
- Dom.ria дозволяє шукати нерухомість за допомогою інтерактивної карти, де можна побачити всі пропозиції в потрібних районах міста;
- На сайті можна знайти каталог перевірених агентств нерухомості та ріелторів, які допоможуть з купівлею або продажем нерухомості.
- Dom.ria має мобільний додаток, який дозволяє користувачам шукати та бронювати нерухомість зручно зі смартфона.

Ці можливості роблять Dom.ria одним з найпопулярніших ресурсів для пошуку нерухомості в Україні. Ви можете дізнатися більше на їхньому офіційному веб-сайті.

Таблиця 1.1 - Порівняння платформ

	Критерій	Lun.ua	Hotels.com	RoomGuru.com	Booking.com
1	Основна функція	Пошук нерухомості для купівлі/оренди	Бронювання житла	Порівняння цін на житло	Бронювання житла
2	Тип об'єктів	Квартири, будинки, новобудови, комерційна нерухомість	Готелі, апартаменти, курорти	Готелі, апартаменти, курорти	Готелі, апартаменти, хостели, будинки
3	Географія покриття	Україна	Світовий ринок	Світовий ринок	Світовий ринок

4	Інтерактивна карта	Так	Так	Так	Так
5	Пошук за ціною	Так	Так	Так	Так
6	Пошук за іншими параметрами	Тип об'єкта, розташування, площа, ціна	Рейтинг, зірковість, ціна, зручності	Тип об'єкта, ціна, рейтинг	Рейтинг, зручності, ціна, локація
7	Система лояльності	Немає	Накопичення безкоштовних ночей	Немає	Genius-програма знижок
8	Агрегація пропозицій	Так	Ні	Так	Ні
9	Монетизація	Реклама, преміум-пропозиції	Комісії за бронювання	Партнерська комісія	Комісії за бронювання
10	Доступ до аналітики	Так (динаміка цін, популярність районів)	Ні	Ні	Ні
11	Основна перевага	Актуальна інформація для ринку України	Програма лояльності	Найкращі ціни через метапошук	Широка база даних і зручність

Продовження табл. 1.1

12	Основний недолік	Вузька спеціалізація на українському ринку	Відсутність агрегованих пропозицій	Обмежена функціональність для бронювання	Висока комісія для власників житла
13	Україномовний інтерфейс	Так	Ні	Так	Так
14	Потрібна реєстрація	Ні	Так	Так	Так

Отже, існуючі аналоги (Lun.ua, Hotels.com, RoomGuru.com, Booking.com) добре підходять для своїх ніш, але не можуть задовольнити усі потреби одночасно (див. табл. 1.1). Наприклад, Lun.ua спеціалізується на ринку нерухомості України, але не підтримує бронювання житла чи пошук поза межами країни. Hotels.com та Booking.com зручні для бронювання житла туристами, але не надають можливостей для купівлі чи аналізу ринку нерухомості. RoomGuru.com добре справляється з порівнянням цін, але не має функціоналу для інших аспектів роботи з нерухомістю.

Жоден з цих сервісів не забезпечує комплексний підхід, який об'єднав би можливості пошуку, порівняння та аналізу нерухомості із зручним інструментом для користувачів у різних географічних регіонах. Це свідчить про необхідність розробки нового веб-застосунку, який може забезпечити універсальність для різних типів нерухомості, інтеграцію з деякими іншими сервісами для порівняння пропозицій тощо.

1.3 Технічне завдання на розробку веб-застосунку для бронювання житлових та нежитлових приміщень

Будь-яка веб-розробка на початковому етапі повинна передбачати створення технічного завдання (ТЗ). Цей документ регламентує функціональність веб-застосунку та визначає вимоги до процесу розробки та звітної документації. При цьому виконавець робіт повинен зрозуміти сутність завдання, показати замовнику приблизний інтерфейс майбутнього веб-продукту, а замовнику зрозуміти, підтвердити та засвідчити, що саме йому потрібно у цьому веб-застосунку.

При написанні ТЗ ми орієнтувалися на загальні принципи розробки програмного забезпечення та врахували специфіку веб-розробки. Загальні визначення:

Повна назва веб-продукту: веб-застосунок для бронювання житлових та нежитлових приміщень.

Замовник – організація, яка замовляє створення веб-застосунку.

Виконавець – розробник, який розробляє веб-застосунок та забезпечує його розгортання.

Next.js – React-фреймворк для створення веб-застосунків з серверним рендерингом.

Supabase – бекенд як сервіс (BaaS), що надає базу даних PostgreSQL, аутентифікацію, зберігання файлів та інші функції.

PostgreSQL – реляційна система управління базами даних. У цьому документі наводиться повний перелік вимог до програмної реалізації та впровадження веб-застосунку. У цьому документі Замовник і Виконавець підтверджують їх згоду з нижченаведеними фактами і умовами:

Виконавець підготував і розробив цей документ, що іменується Технічне Завдання, який містить перелік вимог до виконуваних робіт.

2. Призначення розробки

2.1 Призначення і мета розробки веб-застосунку

Предметом розробки є веб-застосунок для бронювання житлових та нежитлових приміщень.

Веб-застосунок призначений для: відображення доступних для бронювання приміщень з детальною інформацією (фотографії, опис, ціна, зручності), здійснення бронювання приміщень на певний період часу, керування бронюваннями (створення, редагування, скасування), адміністрування приміщень та користувачів.

2.2 Характеристика об'єктів автоматизації

Об'єктами автоматизації є процеси бронювання житлових та нежитлових приміщень, такі як квартири, будинки, офіси, конференц-зали тощо. Веб-застосунок має забезпечити зручний та інтуїтивно зрозумілий інтерфейс для користувачів, а також ефективні інструменти для адміністрування.

3. Вимоги до веб-застосунку

3.1 Вимоги до оформлення інтерфейсу

Графічний інтерфейс веб-застосунку повинен бути сучасним, адаптивним (для різних пристроїв), інтуїтивно зрозумілим та зручним у використанні. Повинні бути передбачені:

Пошук та фільтрація приміщень за різними критеріями (тип, ціна, розташування, зручності).

Календар для вибору дат бронювання.

Особистий кабінет користувача з історією бронювань.

Адміністративна панель для керування приміщеннями, користувачами та бронюваннями.

3.2 Вимоги до зберігання даних і технологій розробки

- Бекенд: Supabase (PostgreSQL).
- Фронтенд: Next.js (React).
- Мова програмування: TypeScript.

3.3 Вимоги до розділення доступу

Веб-застосунок повинен передбачати різні рівні доступу:

- Неавторизований користувач: перегляд доступних приміщень.

- Авторизований користувач: бронювання приміщень, перегляд історії бронювань.
- Адміністратор: повний доступ до керування системою.

3.4 Вимоги до програмного та технічного забезпечення

Для функціонування веб-застосунку необхідний веб-браузер на стороні користувача. Серверна частина розміщується на платформі Supabase.

4. Стадії і етапи розробки веб-застосунку

4.1 Склад і зміст робіт із розробки веб-застосунку

Розробка веб-застосунку для бронювання приміщень передбачає виконання наступних етапів:

- Аналіз предметної області та збір вимог.
- Проектування бази даних.
- Розробка API (Supabase).
- Розробка фронтенду (Next.js).
- Тестування та налагодження.
- Розгортання на платформі Supabase.

5. Порядок контролю і приймання

5.1 Перевірка працездатності роботи веб-застосунку

Приймання веб-застосунку до експлуатації складається з наступної перевірки:

- Перевірка функціональності пошуку та фільтрації.
- Перевірка процесу бронювання.
- Перевірка роботи особистого кабінету користувача.
- Перевірка роботи адміністративної панелі.
- Перевірка адаптивності інтерфейсу.

5.2 Вимоги до введення веб-застосунку в дію

- Підготовка веб-застосунку до експлуатації передбачає:
- Розгортання на платформі Supabase.
- Надання доступу Замовнику до адміністративної панелі.
- Надання документації користувача.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ БРОНЮВАННЯ ЖИТЛОВИХ ПРИМІЩЕНЬ

2.1 Концептуальна модель використання веб-застосунку для бронювання житлових приміщень

Перелік основних скорочень та використовуваних термінів в нашій предметної області представлений у таблиці 2.1.

Таблиця 2.1 Глосарій

Термін	Опис терміну
Програмне забезпечення	сукупність програм системи оброблення інформації та програмних документів, необхідних для експлуатації цих програм
Аккаунт	обліковий запис користувача, яка створюється після реєстрації в будь-якій системі (наприклад, на сайті або в програмі). Як правило, для входу в свій аккаунт вам потрібно ввести логін і пароль.
Алгоритм	послідовність дій для вирішення певної задачі.
Автоматизація	є одним з напрямів науково-технічного прогресу, який спрямовано на застосування саморегульованих технічних засобів, економіко-математичних методів і систем керування, що звільняють людину від участі у процесах
База даних	програмний або програмно-апаратний комплекс, розроблений для зберігання великого обсягу різноманітної інформації.

Продовження табл. 2.1

Інтегрована середовище розробки	це комплекс програмного забезпечення, яке містить все необхідне для розробки, компіляції, лінкінга і налагодження коду.
Командний рядок	різновид текстового інтерфейсу між людиною і комп'ютером, в якому інструкції комп'ютера даються в основному шляхом введення з клавіатури текстових рядків.
Масив	це сукупний тип даних, який дозволяє отримати доступ до всіх змінних одного і того ж типу даних через використання одного ідентифікатора.
Область видимості	визначає, де можна використовувати змінну.
ОС	Операційна Система - комплекс взаємопов'язаних програм, призначених для управління ресурсами комп'ютера та організації взаємодії з користувачем.
Парсинг	збирати (аналізувати) певні дані.
Програма	набір інструкцій, які вказують комп'ютеру, що йому потрібно робити.
Фреймворк	Набір інструментів і бібліотек, які спрощують розробку веб-застосунків
Шаблон	дозволяє написати одну версію функції або класу, яка буде працювати з різними типами даних. Функція або клас, реалізована через шаблон, з фактичним (одним) типом даних називається екземпляром.
Адміністратор	Особа, яка керує веб-застосунком, додаючи нові об'єкти, редагуючи інформацію тощо.

Продовження табл. 2.1

Користувач	Особа, яка використовує веб-застосунок для бронювання
Програмування	наукова і практична діяльність по створення програм
Програмний модуль	программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями для загрузки в оперативную память
Цикл	послідовність команд в програмі, яка повинна виконуватися неодноразово в результаті переходу від початку послідовності до кінця
Буфер	робоча область пам'яті при пересиланні даних
API	Програмовий інтерфейс, який дозволяє різним програмам взаємодіяти між собою
Деплоймент	Розгортання веб-застосунку на виробничому сервері

Етап проектування інформаційних систем є необхідною складовою життєвого циклу програмного забезпечення. Його виконання впливає в кінцевому рахунку на успішність програмного засобу серед кінцевих користувачів. Для проектування програмного забезпечення використовують структурну та об'єктно-орієнтовану методологією проектування [6, с. 226].

Концептуальна модель – це модель предметної області, яка складається з переліку взаємопов'язаних понять, що використовуються для опису цієї області, разом з властивостями й характеристиками, класифікацією цих понять, за типами, ситуаціями, ознаками в даній області і законами протікання процесів у ній. Концептуальна модель визначає структуру модельованої системи, властивості її елементів і причинно-наслідкові зв'язки, що властиві системі і суттєві для досягнення мети моделювання.

Основні елементи концептуальної моделі:

- умови функціонування об'єкта: Визначають характер взаємодії між об'єктом і його оточенням, а також між елементами об'єкта;
- мета дослідження об'єкта: Описує напрями покращення функціонування об'єкта;
- можливості керування об'єктом: Включає визначення складу керованих змінних об'єкта.

Універсальним засобом об'єктно-орієнтованого моделювання є уніфікована мова UML (Unified Modeling Language), розробка якої розпочалася з середини 90-х років минулого століття на базі декількох об'єктноорієнтованих методів і нотацій опису інформаційних систем. Зараз мова UML є загальноприйнятим стандартом документування процесу створення інформаційних систем і програмного забезпечення. На сьогодні існує версія UML 2.0. До пакетів, які працюють з UML, можна віднести IBM Rational Rose, Telelogic TAU G2, Microsoft Visio, Enterprise Architect, Modelio та ін. [23, с. 12]. Мова UML активно застосовується для проектування реляційних БД за допомогою діаграм класів. У межах термінології UML діаграмою класів називається діаграма, на якій показаний набір класів, зв'язків між ними, а також може містити коментарі та обмеження [22, с. 7].

Діаграма варіантів використання є ключовим інструментом візуалізації функціональних вимог до системи (рис. 2.1). Вона ілюструє, як різні типи користувачів (актори) взаємодіють із системою для виконання певних завдань. Для веб-застосунку бронювання житлових приміщень діаграма демонструє всі основні функції, такі як реєстрація, управління обліковими записами, пошук і бронювання житла, здійснення платежів та залишення відгуків. Ця модель допомагає визначити ключові сценарії використання та забезпечує розуміння взаємодії між системою та користувачами.



Рисунок 2.1 – Діаграма варіантів використання

Специфікація варіантів використання є текстовим описом сценаріїв взаємодії користувачів із системою, що доповнює діаграму варіантів використання. Вона деталізує кожен сценарій, включаючи його мету, основний потік подій, альтернативні потоки та виняткові ситуації. Цей документ слугує основою для проектування інтерфейсу користувача та логіки системи, а також допомагає команді розробників і зацікавленим сторонам краще зрозуміти функціональні вимоги системи.

Таблиця 2.2 - Розкадровка варіантів використання.

Варіант використання	Контекст використання	Дійові особи	Передумов а	Тригер
Оформлення бронювання	Клієнт хоче забронювати житло на певний період.	Клієнт, Власник житла, Адмінастратор	Користува ч зареєстров аний; Об'єкт доступний.	Натискання кнопки «Забронювати»

Продовження табл. 2.2

Додавання житлового приміщення	Власник додає новий об'єкт для оренди.	Основна: Власник житла.	Власник авторизований.	Натискання кнопки «Додати об'єкт»
Пошук житлового приміщення	Користувач шукає житло за певними критеріями.	Основна: Клієнт.	Користувач зареєстрований або гість.	Введення пошукових параметрів
Управління бронюваннями	Власник переглядає або змінює бронювання.	Основна: Власник житла.	Власник авторизований, є бронювання.	Перехід у розділ «Мої бронювання».
Зміна профілю	Користувач редагує інформацію у своєму профілі.	Основна: Клієнт або Власник.	Користувач авторизований.	Натискання кнопки «Редагувати профіль».

Окремо зазначимо сценарій та постумови варіантів використання у таблиці 2.3.

Таблиця 2.3 - Розкадровка варіантів використання (сценарій та постумова)

Варіант використання	Сценарій	Постумова
----------------------	----------	-----------

Продовження табл. 2.3

Оформлення бронювання	1. Відкриття сторінки об'єкта. 2. Вибір дат. 3. Перехід до оплати. 4. Підтвердження бронювання.	Інформація збережена, об'єкт оновлено. Додавання житлового приміщення
Додавання житлового приміщення	1. Заповнення форми з описом житла. 2. Завантаження фото. 3. Збереження інформації.	Новий об'єкт додано до бази даних.
Пошук житлового приміщення	1. Введення параметрів. 2. Вибір із результатів.	Показано список об'єктів, що відповідають запиту.
Управління бронюваннями	1. Перегляд бронювань. 2. Зміна статусу або деталей.	Дані бронювання оновлені або скасовані.
Зміна профілю	1. Редагування полів профілю. 2. Підтвердження змін.	Дані профілю оновлені в системі.

Зазначимо елементи керування, що можуть бути створені для реалізації варіантів використання:

- оформлення бронювання: екран вибору житла (список приміщень із кнопкою «Деталі»); деталі житла (інформація про житло, кнопка «Забронювати»); вибір дат (календар); підтвердження (повідомлення про успішне бронювання);

- додавання житла: управління об'єктами (список із кнопкою «Додати об'єкт»); форма додавання (назва, опис, фото, кнопка «Зберегти»); підтвердження (повідомлення про успішне додавання).

- пошук житла: екран пошуку (параметри: місто, дати, ціна, кнопка «Знайти»); результати (список житла з кнопкою «Деталі»).
- управління бронюваннями: список бронювань (кнопки «Переглянути», «Скасувати»); деталі бронювання (інформація про статус, кнопки для змін).
- зміна профілю: екран профілю (дані з кнопкою «Редагувати»); редагування (зміна даних, кнопка «Зберегти»); підтвердження (повідомлення про успішне оновлення).

2.2 Функціональна модель веб-застосунку для бронювання житлових приміщень

Для визначення функціональних вимог до веб-застосунку для бронювання приміщень розробляється функціональна модель з використанням методології IDEF0. Мета цієї моделі — визначити основні функції веб-застосунку, їх взаємозв'язки та інформаційні потоки, а також вхідні і вихідні дані, що використовуються на кожному етапі процесу. Функціональна модель допомагає чітко окреслити структуру веб-застосунку, показати, як різні підсистеми взаємодіють між собою, та визначити роль кожної з них.

Модель IDEF0 дозволяє представити систему як набір взаємопов'язаних функцій, кожна з яких виконує певну задачу в процесі бронювання приміщень. Спочатку система відображається як єдине ціле — контекстна діаграма (A0), що демонструє загальний процес і взаємодію веб-застосунку з іншими системами, такими як користувачі, база даних, адміністрація тощо.

Контекстна діаграма включає кілька ключових елементів: вхід, вихід, управління та механізми. Вхідними даними є інформація, яку користувач надає системі для виконання функцій — це можуть бути дані користувача, параметри пошуку житла (місто, дати, ціна тощо) або запити на бронювання.

Вихідними є результати роботи системи: підтвердження бронювання, список доступних приміщень або інформація про статус бронювання. Управління — це інформація, яка визначає, як і скільки потрібно виконати певних завдань, наприклад, правила для бронювання, перевірка доступності об'єктів на конкретні дати або умови для скасування бронювання. Механізми — це ресурси, які виконують роботу: сервери, бази даних, користувачі інтерфейсу та адміністрація, які забезпечують виконання всіх функцій системи.

Після створення контекстної діаграми, кожен з функціональних блоків системи (наприклад, процес бронювання житла) деталізується на підфункції. Кожна підфункція може бути ще детальніше розбита на окремі блоки для більш глибокого аналізу. Це дозволяє створити чітку картину того, як кожен етап процесу взаємодіє з іншими, які дані використовуються та які ресурси для цього необхідні. Наприклад, блок «Оформлення бронювання» можна поділити на такі підфункції: пошук житла, перегляд деталей обраного житла. Кожна з цих підфункцій може бути також детально розписана на більш дрібніші етапи для визначення конкретних вимог і механізмів їх виконання.

IDEF0 допомагає не лише розробити загальну архітектуру системи, а й деталізувати кожен її частину, щоб зрозуміти, які дані і ресурси використовуються в кожному процесі. В результаті можна отримати функціональну модель, яка надає чітке уявлення про функціонування веб-застосунку для бронювання приміщень, визначає його компоненти та їх взаємодію (рис. 2.2) (рис. 2.3).

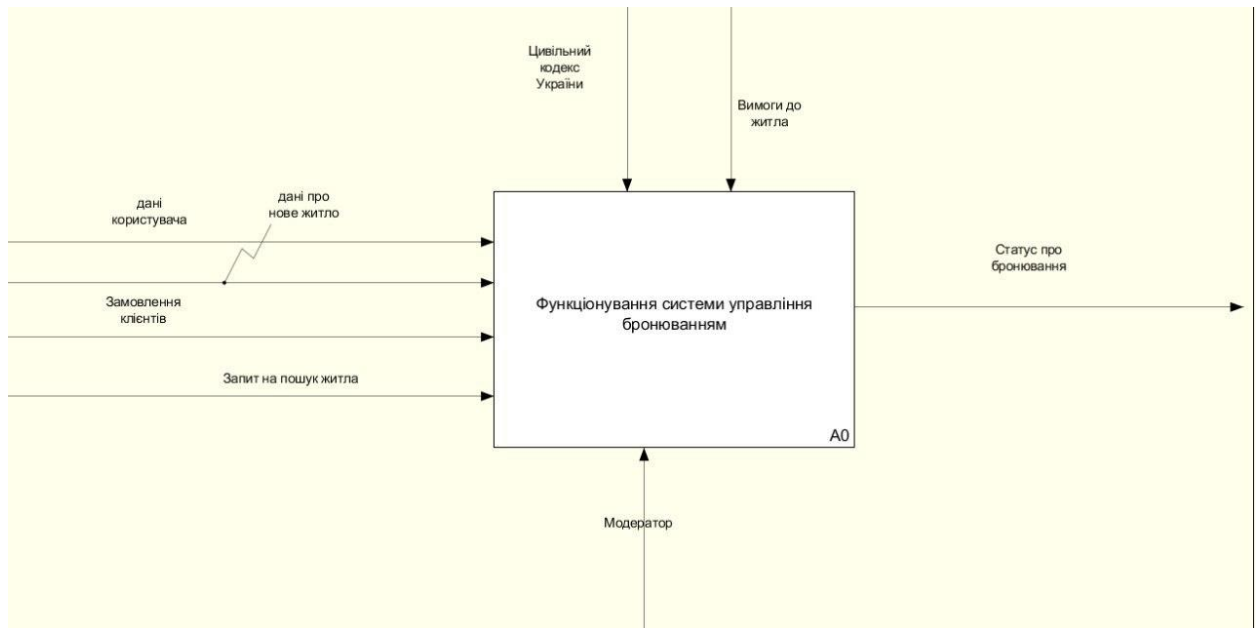


Рисунок 2.2 – Діаграма IDEF0

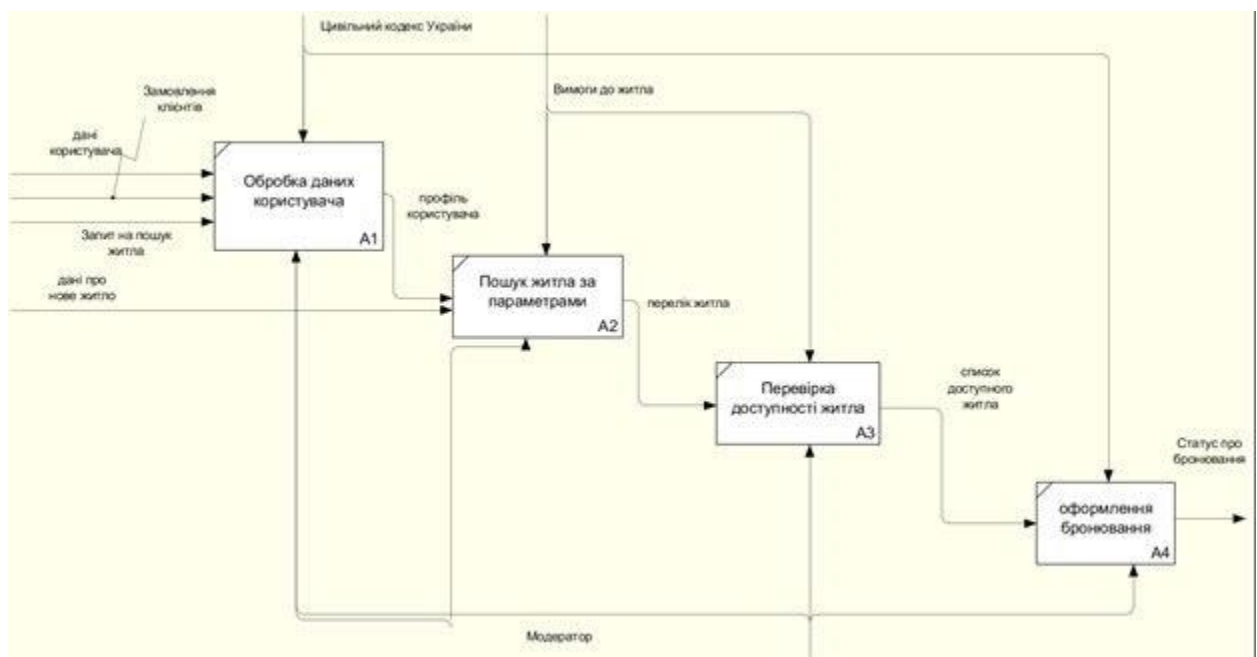


Рисунок 2.3 – Діаграма декомпозиції

2.3. Специфікація функціональних та нефункціональних вимог

Розділ функціональних та нефункціональних вимог є одним із ключових етапів у розробці веб-застосунку для бронювання приміщень. Функціональні вимоги визначають, які конкретні завдання має виконувати

система для задоволення потреб користувачів і адміністраторів. Вони охоплюють усі основні функції, які має реалізувати система, починаючи від реєстрації та аутентифікації користувачів до управління бронюваннями та адмініструванням об'єктів житла. Ці вимоги визначають взаємодію користувача з веб-застосунком та описують, як система повинна обробляти запити, надавати інформацію та підтримувати різноманітні сценарії використання (табл. 2.3).

Таблиця 2.4 - Специфікація функціональних вимог.

	Вимога	Пріоритет	Складність	Контакт
1	Реєстрація та аутентифікація	Обов'язкове	Високе	Користувач
2	Пошук житла за параметрами (місто	Обов'язкове	Середнє	Користувач
3	Перегляд деталей житла (фото	Обов'язкове	Високе	Користувач
4	Оформлення бронювання (підтвердження бронювання)	Обов'язкове	Високе	Користувач
5	Управління бронюваннями (перегляд	Обов'язкове	Середнє	Користувач
6	Додавання нового житла адміністраторами	Рекомендоване	Високе	Модератор
7	Зміна профілю користувача	Рекомендоване	Середнє	Користувач

Продовження табл. 2.4

8	Фільтрація житла за додатковими параметрами (тип об'єкта)	Опційне	Середнє	Користувач
9	Перевірка доступності об'єкта на вказані дати	Обов'язкове	Високе	Користувач
10	Автоматичне підтвердження бронювання через електронну пошту	Рекомендоване	Низьке	Користувач
11	Підтримка різних мов інтерфейсу	Рекомендоване	Середнє	Модератор
12	Інтерфейс для адміністрування бронювань	Опційне	Високе	Модератор

Нефункціональні вимоги, в свою чергу, зосереджуються на характеристиках системи, які забезпечують її ефективну і безпечну роботу. Вони охоплюють аспекти, що визначають якість роботи застосунку. Ці вимоги не описують конкретні функції системи, але вони є невід'ємною частиною її розробки, оскільки гарантують, що система буде працювати стабільно та надійно в умовах реального використання (табл. 2.4).

Визначення чітких функціональних і нефункціональних вимог є необхідним для формування правильної архітектури веб-застосунку, для ефективного управління процесом розробки та тестування, а також для забезпечення якості і стабільності системи після її впровадження. У цьому розділі ми докладно описуємо всі основні функції системи, а також визначаємо важливі нефункціональні аспекти, що впливають на загальний досвід користувачів та ефективність роботи застосунку.

Таблиця 2.5 - Специфікація нефункціональних вимог.

	Назва вимоги	Характеристика
	Застосовність	
1.	Час для навчання користувачів	Час на освоєння основних функцій застосунку — до 1 години.
2.	Вимірюваний час відгуку для типових завдань	Час відгуку не більше 2 секунд для основних функцій (пошук, бронювання).

Продовження табл. 2.5

	Надійність	
1	Середній час безвідмовної роботи	Система повинна працювати без збоїв 99.9% часу.
2	Середнє напруження до ремонту	Мінімум 2000 годин без збоїв до необхідності ремонту.
	Робочі характеристики	
1	Швидкодія для транзакції	Час для здійснення однієї транзакції не більше 3 секунд.
2	Продуктивність	Система повинна обробляти до 1000 одночасних користувачів.
	Проектні обмеження	
1	Мова програмування	Використання JavaScript, HTML, CSS для фронтенду; Python/Node.js для бекенду.
2	Платформа	Підтримка сучасних веб-браузерів (Chrome, Firefox, Safari, Edge).
	Вимоги до документації, призначеної для користувача, і до системи допомоги	

1	Документація для користувачів	Доступна інтерактивна документація, що допомагає користувачам освоїти систему.
2	Система допомоги	Вбудоване FAQ та чат з підтримкою для користувачів.
	Інтерфейси	
1	Інтерфейси користувача	Інтуїтивно зрозумілий інтерфейс, підтримка адаптивного дизайну для всіх пристроїв.

Продовження табл. 2.5

2	Програмні інтерфейси	API для інтеграції з іншими системами (наприклад, платіжними шлюзами).
3	Комунікаційні інтерфейси	Використання стандартів HTTPS для безпечної комунікації.
	Вимоги до ліцензування	
1	Ліцензія на програмне забезпечення	Використання ліцензії Open Source для більшості компонентів, ліцензія на платні сервіси за потребою.
	Вживані стандарти	
1	Стандарти програмування	Дотримання сучасних стандартів кодування (ESLint, PEP8 для Python).
2	Стандарти безпеки	Використання стандартів OWASP для захисту від уразливостей.

РОЗДІЛ 3 ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ ВЕБ-ЗАСТОСУНКУ ДЛЯ БРОНЮВАННЯ ЖИТЛОВИХ ПРИМІЩЕНЬ

3.1 Загальний огляд технологічного стеку

Більшість сайтів з бронювання пропонують стандартні форми пошуку житла та його бронювання. До основних параметрів пошуку належать рядок пошуку, де користувач вводить місто або населений пункт, дату заїзду та дату від'їзду, кількість осіб, ціновий діапазон та тип розміщення. По заданому місту можна отримати перелік усіх варіантів розміщення, що є у місті на зразок кімнат, квартир, будинків або готелей. Дата заїзду та від'їзду дають можливість задати конкретний або приблизний діапазон часу перебування. В такому випадку система надасть перелік вільного житла у зазначений період. Ціновий діапазон дає можливість переглянути результати пошуку з житлом, вартість якого не перевищуватиме певної суми, таким чином підібрати найбільш економний варіант. Вказуючи кількість осіб можна знайти усі варіанти розміщення на певну кількість осіб. [2], [3]. Кожний результат пошуку має детальний опис житла та форму для його бронювання. У результаті бронювання формується документ про здійснення броні, що висилається користувачеві [20, с. 73].

Наявність web-сайту для будь-якого бізнесу є ознакою сучасності, конкурентоспроможності та можливістю вплинути на економічний ріст компанії. У нашому випадку web-сайт сервісу бронювання об'єднує в собі плюси розвитку бізнесу та підвищену зручність пошуку житла для клієнтів. Такий сервіс сприяє тісному контакту відвідувача з готелем, надаючи рейтинги та відгуки щодо номерів та апартаментів. Платформа для бронювання має багато допоміжних функцій та категорій для фільтрування вимог для користувача на одному сайті, у іншому разі клієнту довелося би годинами вивчати Інтернет простір для пошуку необхідного готелю, який би задовольняв усі потреби потенційного клієнта. У висновку, зрозуміло, що

готельний ринок України відновлюється та вже повноцінно функціонує у відносно безпечних регіонах, а web-сайт сервісу бронювання п'ятизіркових готелів та люкс апартаментів стане дієвим помічником для розвитку готельної діяльності та зручності бронювання комфортного сегменту житла в Україні [26, с. 27].

Для створення веб-ресурсів виділяють наступні засоби розробки [8], [10].

- ручна розробка за допомогою HTML – найпоширеніший метод створення веб-сторінок з появою стандарту HTML. Основним інструментом для створення є прості текстові редактор, а також візуальних текстових редакторах HTML та CSS в режимі WYSIWYG, тобто «Що бачу, те й отримую». Для створення статичного сайту з використанням HTML, CSS та JavaScript цілком достатньо. Проте, за допомогою такого способу зробити динамічний сайт досить складно, оскільки потрібно буде використовувати серверні скрипти, які будуть потребувати додаткового встановлення програмного забезпечення для роботи з ними. Такий метод має незаперечну перевагу, оскільки при такому методі створення отримує саме такий сайт, який планували, тому багато веброзробників надають перевагу саме такому способу створення;

- за допомогою програмних засобів розробки сайтів – це візуальні редактори HTML-коду, які використовують для спрощення розробки веб-сторінок, оскільки код сторінки може містити велику кількість рядків. В таких редакторах застосовують готові шаблони оформлення та можливість генерувати HTML-код. Виділяють програми, які у своєму складі містять візуальні редактори (designbased editor), які автоматично генерують необхідний та дозволяють працювати в режимі WYSIWYG (приклад найбільш популярних редакторів: Adobe DreamWeaver, Microsoft FrontPage тощо) та програми редактори (code-based editor), які надають допоміжні засоби для автоматичного створення коду (приклад найбільш популярних

редакторі: Adobe HomeSites, HotDog, AceHTML, Visual Studio Code, Sublime Text, PhpStorm, Atom, Brackets тощо).;

- за допомогою інструментальних систем (CMS) – це готова візуальна програмна оболонка, яку потрібно заповнити контентом та за потреби знести зміни в її налаштування. Фактично, це система керування контентом. Найбільш популярний метод створення, оскільки є зручним та практичним засобом, що економить час та сили на створення коду, оскільки не потребує спеціальних навичок веб-програмування. Особливість роботи з даною системою полягає в тому, що контент цілком є відокремлений від візуального представлення сайту, Секція 1. Інформаційно-комунікаційні технології в освіті та науці 190 таким чином полегшується процес редагування його вмісту. Найбільш популярними CMS є WordPress, Joomla, Drupal тощо;

- з використанням фреймворків – це каркас або основа, який використовують для створення динамічних веб-сайтів. Містить завчасно створенні певні реалізовані класи та функції. Тобто, до певної задачі можна добавляти свій код, до вже створеного на фреймворку. Застосування даного засобу звільняє від необхідності писати рутинний код та спрощує доступ до бази даних та розробки інтерфейсу, що зменшує дублювання коду. Найбільш популярними фреймворками є Bootstrap, Foundation, Uikit, Django, Tornado тощо;

- на базі SaaS-платформ у хмарному середовищі Cloud – спеціальні конструктори, що дозволяють в онлайн-режимі зібрати сайт на основі готових модулів та компонентів. Використовуються для простих веб-проектів з мінімальними вимогами до знань розробника та за досить швидко. Перевагою даного засобу є те, що не потрібно турбуватись над розробкою проекту, встановленням необхідних програм, розміщені на хостингу та його налаштуваннями, також в подальшому не потрібно займатись його супроводом. Проте мінусом даного способу є шаблонний дизайн та неможливість вийти за рамки функціональних можливостей. Найбільш популярними SaaS-платформ в Cloud є UMI, WIX, InSales, Shopify та сайти,

які надають послуги адміністрування сайту sites.google.com, weedly.com, webnode.com.ua, ucoz.ua тощо.

Next.js – це фреймворк, розроблений Vercel, відкритий вихідний код, заснований на Node.js та Babel, який інтегрується з React для розробки односторінкових програм. Без перебільшень Next можна назвати молодшим братом React-бібліотеки, адже він повністю реалізує його функціонал, а також додає ряд особливостей, роблячи його більш легким та швидким у використанні. Переваги Next.js:

- Server-side-render (далі SSR) – додатковий інструмент для взаємодії із серверною частиною проекту. Він дозволяє отримати та налаштувати доступ до всіх необхідних даних на стороні сервера, у тому числі і до живого сервера, коли ви бачите зміни коду на сторінці в режимі реального часу;
- Static-site-generation (далі SSG) – інструмент, що дозволяє попередньо підвантажувати веб-сторінки, тим самим скорочуючи час завантаження та покращуючи продуктивність веб-додатку;
- поліпшено пошукову оптимізацію проектів. Частково це заслуга SSR і SSG, оскільки якісніша робота з backend дає перевагу в SEO-оптимізації. Пошукові системи краще ранжують більш чуйні сайти, а також вони сприяють більш швидшому завантаженню сторінок.;
- можливість редагування мета-тег. У React.js цей тег не редагувався, але це найважливіший мета-компонент веб-сайту, який допомагає просувати вебсайт у результатах пошуку;
- Next.js має вбудовану обробку зображень, що спрощує роботу з графікою та забезпечує більш швидке завантаження зображень на сайті. За допомогою вбудованих функцій можна змінювати розмір та формат зображень, що також сприяє покращенню продуктивності;
- інтуїтивний маршрутизатор, який дозволяє розробникам легко керувати маршрутами та створювати динамічні сторінки. Це спрощує процес розробки та робить додатки більш гнучкими. Додатки, створені з Next.js допомогою, працюють неймовірно швидко, їх легко розгорнути, а також

легко налаштувати. Цей фреймворк ідеально підходить для створення лендінг сторінок. Частина маркетологів використовують його для створення цільових або інших сторінок, орієнтованих на маркетинг. Конкурентною перевагою Next.js є SEO-оптимізація веб-додатків, тому це має важливе значення при створенні інтернет-магазинів або ін. Створення рекламних веб-сайтів. Оскільки програма, написана на Next.js використовує серверний рендеринг (SSR), розробник може скоротити час завантаження, в першу чергу, коли відвідувачі використовують повільніші пристрої (рис. 3.1).

Отже, Next.js – це найкращий вибір для збільшення органічного трафіку завдяки його швидкості, рендерингу на стороні сервера та унікальним функціям [16].

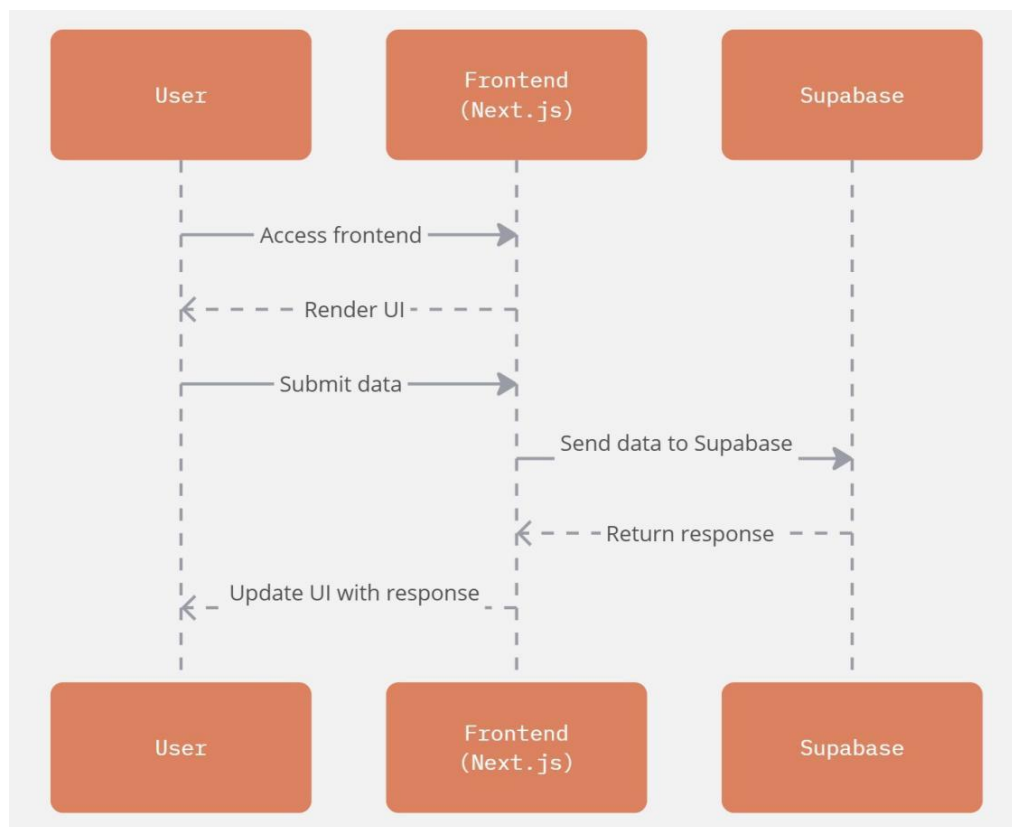


Рисунок 3.1 - Структурна схема додатку

3.2 Обґрунтування вибору мови програмування

У сучасному світі веб-технологій вибір мови програмування для створення веб-застосунків є критично важливим етапом, що безпосередньо впливає на ефективність, продуктивність і масштабованість проекту. Особливо це стосується сфери оренди приміщень, де швидкість обробки запитів, зручність користувацького інтерфейсу та безпека даних грають ключову роль у забезпеченні позитивного досвіду користувачів. Веб-застосунки для оренди приміщень повинні бути здатні обробляти великий обсяг інформації, включаючи деталі про приміщення, ціни, наявність та відгуки користувачів. Крім того, важливо забезпечити інтеграцію з платіжними системами та можливість швидкого пошуку і фільтрації пропозицій. У цьому контексті вибір мови програмування та фреймворку може суттєво вплинути на реалізацію цих функцій.

У даному підрозділі ми розглянемо ключові критерії, які слід враховувати при виборі мови програмування для веб-застосунку в сфері оренди приміщень. Ми також проаналізуємо різні технології, їх переваги та недоліки, а також обґрунтуємо вибір конкретних рішень, що найкраще відповідають потребам проекту. Це дозволить створити ефективний, зручний і безпечний продукт, що задовольняє вимоги як бізнесу, так і кінцевих користувачів.

У сучасній розробці веб-додатків JavaScript (JS) залишається основною мовою програмування для фронтенду, завдяки своїй універсальності та широкій підтримці в браузерях. Однак, з розвитком технологій та зростанням вимог до якості коду, TypeScript (TS) став все більш популярним вибором серед розробників. Поєднання TypeScript та JavaScript створює потужну платформу для розробки фронтенд-додатків, що має кілька важливих переваг.

Вибір TypeScript для фронтенд-розробки є логічним кроком у сучасному контексті веб-технологій. Завдяки своїм перевагам, він забезпечує не лише високу якість коду, але й покращує продуктивність команди розробників. Використання TypeScript дозволяє створювати більш стабільні

та надійні веб-додатки, що відповідають сучасним вимогам користувачів. Таким чином, TypeScript та JavaScript становлять єдиний і найкращий вибір для фронтенду, оскільки їх поєднання забезпечує оптимальний баланс між продуктивністю, якістю коду та зручністю розробки. Це робить їх ідеальним рішенням для проектів різного масштабу, включаючи складні веб-застосунки в сфері оренди приміщень та інших галузях [4, с. 162–169].

Вибір TypeScript у поєднанні з Next.js для розробки веб-додатків обґрунтований кількома ключовими аспектами, які підвищують продуктивність, якість коду та загальний досвід розробника [33].

Переваги використання TypeScript:

- статична типізація: TypeScript є надмножиною JavaScript, що додає статичну типізацію. Це дозволяє виявляти помилки на етапі компіляції, що значно знижує ймовірність помилок у коді¹³. Статична типізація забезпечує більш чітке визначення типів даних, що підвищує зрозумілість і підтримуваність коду;
- покращена якість коду: Використання TypeScript допомагає уникати поширених помилок, таких як неправильне використання значень або невірні припущення про поведінку коду. Це особливо важливо в великих проектах, де складність коду може призводити до труднощів у підтримці;
- підтримка IDE: TypeScript забезпечує покращену інтеграцію з середовищами розробки (IDE), надаючи функції автозавершення, перевірки типів і рефакторингу, що робить процес розробки більш ефективним.

Переваги Next.js:

- серверний рендеринг (SSR): Next.js дозволяє виконувати серверний рендеринг, що покращує швидкість завантаження сторінок і SEO-оптимізацію. Це особливо корисно для додатків, які потребують швидкого відображення контенту;
- автоматичне розділення коду: Next.js автоматично розділяє код на менші частини, що зменшує обсяг даних, які потрібно завантажити при

первинному відкритті сторінки. Це сприяє покращенню продуктивності додатків;

- легкість у налаштуванні: Завдяки вбудованій підтримці TypeScript у Next.js, налаштування проекту стає простим і швидким. Розробники можуть легко створити проект з підтримкою TypeScript за допомогою команди `create-next-app`.

Комбінація TypeScript та Next.js дозволяє досягти високої продуктивності та стабільності в розробці веб-додатків. Переваги статичної типізації TypeScript доповнюються можливостями Next.js у створенні швидких і SEO-дружніх додатків. Це робить їх ідеальним вибором для сучасних веб-проектів, де важливі якості коду та швидкість завантаження [25].

Існує кілька альтернатив Next.js, які можуть бути використані для розробки веб-додатків, проте кожна з них має свої недоліки, які можуть обмежити їхню придатність для певних проектів [27].

По-перше, Gatsby є популярною альтернативою, яка спеціалізується на статичній генерації сайтів. Хоча Gatsby пропонує чудову продуктивність і SEO-оптимізацію, його фокус на статичному контенті може бути обмеженням для проектів, які потребують динамічного рендерингу або частих оновлень даних. Це робить Gatsby менш придатним для застосунків з високою динамікою контенту.

Angular також часто розглядається як альтернатива Next.js. Він підходить для великих корпоративних додатків завдяки своїй структурі та потужним можливостям. Проте Angular може бути складнішим у вивченні та налаштуванні, що може затримати процес розробки, особливо для невеликих проектів або команд з обмеженим досвідом.

Nuxt.js, який є аналогом Next.js для Vue.js, пропонує подібні функції, але якщо команда вже працює з React, перехід на Nuxt.js може вимагати додаткових зусиль і часу на навчання новій технології.

Remix є ще однією альтернативою, яка акцентує увагу на швидкості та простоті. Однак вона може не мати такої ж гнучкості в маршрутизації та управлінні даними, як Next.js, що може стати проблемою для більш складних проектів.

Нарешті, Svelte і Astro також пропонують цікаві можливості для розробки веб-додатків. Проте Svelte все ще залишається новим у порівнянні з React і Next.js, і його екосистема може бути менш зрілою. Astro ж більше орієнтований на статичні сайти і не підходить для проектів із динамічним контентом.

Таким чином, хоча існують різноманітні альтернативи Next.js, їх недоліки можуть обмежити їхню ефективність у певних сценаріях. Next.js залишається одним із найкращих виборів для створення сучасних веб-додатків завдяки своїй гнучкості, підтримці SSR та простоті інтеграції з React.

Twitch використовує Next.js для забезпечення швидкого завантаження сторінок та покращення взаємодії з користувачами, що є критично важливим для платформи, яка обслуговує мільйони активних користувачів щодня. Використання Next.js дозволяє Twitch реалізувати серверний рендеринг (SSR), що значно покращує продуктивність та SEO-оптимізацію. Next.js підтримує динамічну маршрутизацію і легку інтеграцію з API, що дозволяє Twitch швидко реагувати на запити користувачів і надавати їм актуальний контент. Це особливо важливо для стрімінгових платформ, де швидкість завантаження і доступність контенту є ключовими факторами для утримання аудиторії. Крім того, Next.js забезпечує можливості для оптимізації зображень та інших медіа-ресурсів, що також сприяє покращенню загального досвіду користувачів на платформі. Завдяки цим характеристикам, Twitch може безперешкодно обслуговувати величезні обсяги трафіку та пропонувати користувачам інтуїтивно зрозумілий інтерфейс. Таким чином, використання Next.js на Twitch є прикладом того, як сучасні технології

можуть бути ефективно застосовані для створення масштабованих і продуктивних веб-додатків у сфері стрімінгу.

Supabase є потужною платформою для розробки бекенду, що позиціонується як альтернатива Firebase. Вона пропонує широкий спектр функцій, які роблять її привабливим вибором для розробників, які прагнуть швидко та ефективно створювати веб-додатки. Основні переваги Supabase включають простоту налаштування, потужну базу даних PostgreSQL, реальний час оновлень даних та зручну систему авторизації [40].

Переваги Supabase:

- простота налаштування: Supabase забезпечує функцію «Zero Configuration», що дозволяє розробникам швидко розпочати роботу без необхідності налаштовувати сервери або управляти складними інфраструктурами. Це особливо вигідно для стартапів і незалежних розробників, які хочуть зосередитися на фронтенді.
- потужна база даних PostgreSQL: Supabase використовує PostgreSQL, одну з найпотужніших систем управління базами даних. Це дозволяє легко управляти даними, виконувати складні запити та використовувати розширення PostgreSQL для покращення функціональності;
- реальний час оновлень: Платформа підтримує функцію реального часу, що дозволяє підписуватися на зміни в базі даних і отримувати оновлення миттєво. Це відкриває можливості для створення інтерактивних додатків без необхідності впроваджувати складну логіку;
- зручна система авторизації: Supabase пропонує просту у використанні систему авторизації, яка підтримує різні методи входу, включаючи електронну пошту та паролі, а також OAuth-провайдери. Це забезпечує гнучкість у реалізації авторизації для різних платформ;
- серверні функції: Supabase також підтримує серверні функції на базі Deno, що дозволяє швидко розробляти та розгортати серверну логіку без необхідності управляти інфраструктурою.

Недоліки Supabase:

- відсутність підтримки власних доменів: Одним із обмежень Supabase є те, що проекти доступні лише за доменами на кшталт `yourproject.supabase.com`. Це може бути недоліком для тих, хто хоче використовувати власний домен;
- автоматичне вимкнення бекенду: Якщо проект не має активності протягом певного часу, Supabase автоматично вимикає бекенд. Це може призвести до незручностей, оскільки потрібно вручну перезапустити проект через адміністративну панель;
- необхідність знань PostgreSQL: Хоча Supabase спрощує управління базами даних, розробники повинні мати певне розуміння PostgreSQL для ефективного використання платформи.

Вибір TypeScript разом із Next.js є стратегічно обґрунтованим рішенням для розробників, які прагнуть створювати масштабовані, продуктивні та легко підтримувані веб-додатки. Ця комбінація забезпечує не лише покращену якість коду, але й зручність у роботі з проектами різного масштабу. Supabase є потужним рішенням для бекенду завдяки своїй простоті використання, потужній базі даних та реальному часу оновлень. Хоча платформа має деякі обмеження, її переваги значно перевищують недоліки, роблячи її ідеальним вибором для сучасних веб-додатків у сфері оренди приміщень та інших галузях.

3.3 Інструменти для розробки та CI/CD

Для розробки та тестування застосунків на базі Next.js використовуються різноманітні інструменти, які забезпечують ефективність, зручність та якість коду. Основні категорії інструментів включають IDE, системи управління версіями та бібліотеки для тестування. Одними з найпопулярніших інтегрованих середовищ розробки (IDE) для Next.js є Visual Studio Code та WebStorm.

Visual Studio Code – це безкоштовний редактор коду з потужними функціями, такими як IntelliSense, підтримка Git, а також можливість інтеграції з численними розширеннями, такими як ESLint і Prettier. Це робить його ідеальним вибором для розробників, оскільки він легкий у використанні і має велику спільноту підтримки.

WebStorm, з іншого боку, є платним IDE від JetBrains, який пропонує розширені можливості для JavaScript-розробки, включаючи інтелектуальне автозаповнення коду та потужні інструменти для налагодження. Хоча WebStorm є платним продуктом, його функціональність може виправдати витрати для великих проектів або команд.

Для управління версіями коду зазвичай використовують Git разом із платформами на кшталт GitHub, GitLab або Bitbucket. Ці системи дозволяють командам ефективно співпрацювати, відстежувати зміни в коді та управляти різними версіями проекту. Використання системи контролю версій допомагає уникнути плутанини при внесенні змін і дозволяє легко повертатися до попередніх версій коду [42].

Для тестування застосунків на Next.js популярними є кілька бібліотек і фреймворків. Jest – це один із найвідоміших фреймворків для тестування JavaScript-додатків. Він забезпечує простий інтерфейс для написання юніт-тестів і інтеграційних тестів. У поєднанні з React Testing Library, Jest дозволяє писати тести, які імітують поведінку користувачів, що підвищує якість тестування компонентів. Також варто згадати Cypress та Playwright, які використовуються для енд-ту-енд тестування. Cypress надає інтуїтивно зрозумілий інтерфейс для автоматизації браузерних тестів, що дозволяє перевіряти функціональність додатка з точки зору користувача. Playwright забезпечує можливості крос-браузерного тестування, що є важливим для перевірки сумісності застосунків на різних платформах [41].

Використання цих інструментів у поєднанні дозволяє розробникам створювати високоякісні веб-додатки на базі Next.js, які відповідають сучасним вимогам до продуктивності та стабільності.

Автоматизація розгортання та тестування є ключовими компонентами сучасної розробки програмного забезпечення, особливо у проектах на базі Next.js. Впровадження процесів Continuous Integration (CI) та Continuous Deployment (CD) дозволяє розробникам автоматизувати тестування коду, його збірку та розгортання, що значно підвищує ефективність роботи команди [3, с. 402-403].

Одним із найпопулярніших інструментів для CI/CD є GitLab CI. Налаштування CI/CD в GitLab передбачає створення файлу `.gitlab-ci.yml`, в якому визначаються етапи та завдання для автоматизації. Наприклад, ви можете налаштувати етапи для публікації та розгортання, які будуть виконуватися при кожному коміті в репозиторій. Це дозволяє автоматично тестувати код і розгорнути його на сервері, якщо всі тести пройшли успішно. Іншим популярним рішенням є GitHub Actions, яке інтегрується безпосередньо з репозиторіями на GitHub. Це дозволяє створювати гнучкі робочі процеси для автоматизації тестування, збірки та розгортання проектів. GitHub Actions підтримує різноманітні тригери, такі як пуші або запити на злиття, що робить його дуже зручним для команд, які вже використовують Git для контролю версій.

Також варто згадати Jenkins, який є потужним сервером автоматизації з відкритим вихідним кодом. Jenkins дозволяє налаштовувати складні пайплайни і інтегрується з багатьма іншими інструментами. Це робить його ідеальним для великих проектів, де потрібна велика гнучкість у налаштуванні процесів CI/CD. Для тестування в рамках CI/CD можна використовувати бібліотеки, такі як Jest і React Testing Library, які дозволяють автоматизувати юніт-тестування та інтеграційне тестування компонентів Next.js. Це забезпечує високий рівень впевненості в якості коду перед його розгортанням.

Загалом, впровадження CI/CD у проекти на базі Next.js значно спрощує процеси тестування та розгортання, дозволяючи командам швидко реагувати на зміни в коді і забезпечувати стабільність продукту.

РОЗДІЛ 4 ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ

4.1 Особливості програмної реалізації

Створення сучасного веб-застосунку для оренди приміщень вимагає не лише продуманого дизайну та функціоналу, але й чіткої організації коду, ефективного використання технологій та інструментів. На відміну від традиційних підходів, де серверна частина реалізується за допомогою PHP та MySQL, сучасні рішення на базі Next.js та Supabase пропонують більшу гнучкість, масштабованість та зручність для розробників і користувачів.

Цей розділ описує фізичну структуру проекту, пояснює логіку розподілу файлів і каталогів, а також обґрунтовує вибір технологічного стеку. Чітке структурування коду дозволяє легко масштабувати застосунок, додавати нові функції та підтримувати стабільність системи. Використання Next.js забезпечує швидкість завантаження сторінок і SEO-оптимізацію, а Supabase надає готові інструменти для роботи з базою даних, автентифікацією та файлами, що значно прискорює розробку [30].

Фізично веб-застосунок для оренди приміщень є сучасною веб-платформою, побудованою на базі технологій Next.js (фронтенд) та Supabase (бекенд). Структура проекту організована у вигляді кореневого каталогу з підкаталогами та файлами, що забезпечують функціональність та логіку застосунку:

```

├── pages/           # Сторінки застосунку (Next.js)
|   ├── index.tsx    # Головна сторінка зі списком приміщень
|   ├── listings/    # Сторінки для перегляду та управління оголошеннями
|   |   ├── [id].tsx  # Динамічна сторінка конкретного оголошення
|   |   └── dashboard.tsx # Особистий кабінет користувача
|
├── components/      # React-компоненти (кнопки, картки оголошень,
форми)

```

```

|   |— ListingCard.tsx
|   |— SearchFilters.tsx
|
|— public/          # Статичні ресурси
|   |— images/      # Зображення інтерфейсу (лого, іконки)
|   |— listings/    # Фото приміщень для оренди
|   |— fonts/       # Кастомні шрифти
|
|— styles/          # Глобальні стилі та CSS-модулі
|   |— globals.css  # Головні стилі
|   |— Home.module.css
|
|— lib/             # Клієнтська логіка для роботи з Supabase
|   |— supabaseClient.ts # Ініціалізація клієнта Supabase
|
|— utils/           # Допоміжні функції (форматування дат, валідація)
|   |— dateFormatter.ts
|
|— types/           # Типи TypeScript для оголошень, користувачів тощо
|   |— listing.ts

```

Для реалізації бекенду використовується Supabase, що надає:

- аутентифікацію: інтеграція з OAuth, email/password, SMS;
- базу даних: PostgreSQL зі вбудованим REST/GraphQL API;
- реальний час: підписка на зміни в базі даних (наприклад, оновлення статусу оренди);
- сховище файлів: для зберігання фото приміщень.

Проектування бази даних для веб-застосунку для бронювання житлових приміщень включає кілька основних аспектів: користувачі, житлові приміщення, бронювання, оплата, а також інші можливі

функціональні можливості, такі як відгуки або спеціальні пропозиції. Ось основна структура таблиць, які можна використовувати:

1. Таблиця користувачів (Users). Ця таблиця зберігає дані про користувачів, як звичайних, так і адміністраторів, складається з наступних полів:

- user_id (INT, PRIMARY KEY)
- first_name (VARCHAR)
- last_name (VARCHAR)
- email (VARCHAR, UNIQUE)
- password_hash (VARCHAR)
- phone_number (VARCHAR, NULLABLE)
- role (ENUM: 'user', 'admin')
- created_at (TIMESTAMP)
- updated_at (TIMESTAMP)

2. Таблиця житлових приміщень (Properties). Таблиця для зберігання інформації про житлові приміщення, що здаються в оренду, складається з наступних полів:

- property_id (INT, PRIMARY KEY)
- owner_id (INT, FOREIGN KEY -> Users.user_id)
- title (VARCHAR)
- description (TEXT)
- address (VARCHAR)
- price_per_night (DECIMAL)
- num_bedrooms (INT)
- num_bathrooms (INT)
- max_guests (INT)
- available_from (DATE)
- available_to (DATE)
- created_at (TIMESTAMP)
- updated_at (TIMESTAMP)

3. Таблиця бронювань (Bookings). Зберігає дані про кожне бронювання, включаючи період і статус, , складається з наступних полів:

- booking_id (INT, PRIMARY KEY)
- user_id (INT, FOREIGN KEY -> Users.user_id)
- property_id (INT, FOREIGN KEY -> Properties.property_id)
- check_in_date (DATE)
- check_out_date (DATE)
- total_price (DECIMAL)
- status (ENUM: 'pending', 'confirmed', 'completed', 'canceled')
- created_at (TIMESTAMP)
- updated_at (TIMESTAMP)

4. Таблиця оплат (Payments). Зберігає інформацію про оплати за бронювання, складається з наступних полів:

- payment_id (INT, PRIMARY KEY)
- booking_id (INT, FOREIGN KEY -> Bookings.booking_id)
- amount (DECIMAL)
- payment_method (ENUM: 'credit_card', 'paypal', 'bank_transfer')
- payment_status (ENUM: 'pending', 'completed', 'failed')
- payment_date (TIMESTAMP)

5. Таблиця відгуків (Reviews). Містить відгуки користувачів про житлові приміщення, складається з наступних полів:

- review_id (INT, PRIMARY KEY)
- user_id (INT, FOREIGN KEY -> Users.user_id)
- property_id (INT, FOREIGN KEY -> Properties.property_id)
- rating (INT, 1–5)
- comment (TEXT)
- created_at (TIMESTAMP)

Первинні ключі (PRIMARY KEY) є критично важливими для підтримки цілісності даних у PostgreSQL. Вони забезпечують унікальність записів і запобігають появі дублікатів, що робить їх незамінними в

реляційних базах даних. Правильне використання первинних ключів сприяє ефективному управлінню даними та оптимізації запитів у базі даних [34].

Для реалізації авторизації в веб-застосунку на базі Next.js за допомогою Supabase, вам потрібно виконати кілька кроків, включаючи налаштування проекту, створення клієнтів Supabase, а також реалізацію функцій для входу та реєстрації користувачів [35].

```
// utils/supabaseClient.ts
```

```
import { createClient } from '@supabase/supabase-js';
```

```
const supabaseUrl = process.env.NEXT_PUBLIC_SUPABASE_URL!;
```

```
const supabaseAnonKey =
```

```
process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!;
```

```
export const supabase = createClient(supabaseUrl, supabaseAnonKey);
```

```
// pages/login.tsx
```

```
import { useState } from 'react';
```

```
import { supabase } from '../utils/supabaseClient';
```

```
import { useRouter } from 'next/router';
```

```
const LoginPage = () => {
```

```
  const [email, setEmail] = useState('');
```

```
  const [password, setPassword] = useState('');
```

```
  const router = useRouter();
```

```
  const handleLogin = async (e: React.FormEvent) => {
```

```
    e.preventDefault();
```

```
    const { error } = await supabase.auth.signInWithPassword({ email, password
```

```
  });
```

```
    if (error) {
```

```
      alert(error.message);
```

```
    } else {
```

```
      router.push('/'); // Перенаправлення на головну сторінку після
```

```
успішного входу
```

```
    }  };
```

```

return (
  <form onSubmit={handleLogin}>
    <label>
      Email:
      <input type="email" value={email} onChange={(e) =>
setEmail(e.target.value)} required />
    </label>
    <label>
      Password:
      <input type="password" value={password} onChange={(e) =>
setPassword(e.target.value)} required />
    </label>
    <button type="submit">Log In</button>
  </form>
);};
export default LoginPage;

```

Цей приклад демонструє базову реалізацію авторизації в Next.js за допомогою Supabase. Ви можете розширити цю реалізацію, додаючи функціональність для скидання пароля, підтвердження електронної пошти та управління користувачами. Supabase надає потужний бекенд для управління автентифікацією та дозволяє швидко інтегрувати ці можливості у ваш веб-застосунок.

4.2 Інструкція користувача веб-застосунку для оренди приміщень

Ця інструкція призначена для допомоги користувачам у навігації по веб-застосунку для оренди приміщень. Вона охоплює процеси входу в систему та вибору приміщень на інтерактивній карті.

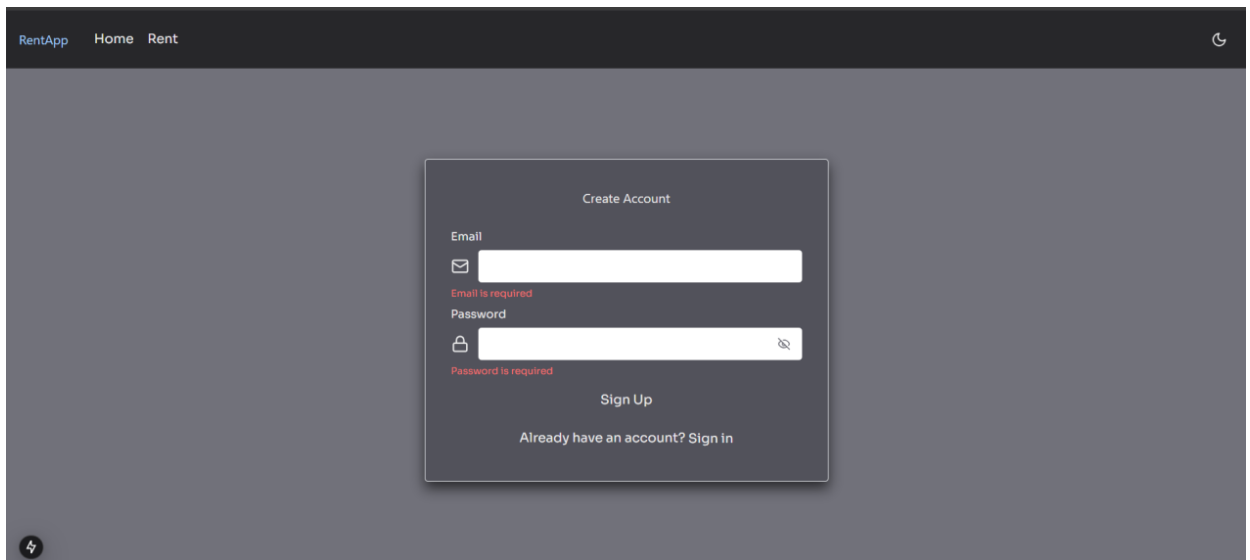


Рисунок 4.1 - Авторизація

Для того щоб отримати доступ до функціоналу веб-застосунку, вам потрібно виконати вхід у систему. Спочатку відкрийте веб-браузер і перейдіть за адресою веб-застосунку. На головній сторінці ви знайдете кнопку «Увійти» або «Логін», яка дозволить вам перейти на сторінку авторизації (рис. 4.1).

На сторінці входу вам буде запропоновано заповнити форму. У полі «Email» введіть вашу електронну пошту, з якою ви зареєструвалися. У полі «Пароль» введіть ваш пароль. Після заповнення всіх полів натисніть кнопку «Увійти». Якщо ви ввели правильні дані, ви будете перенаправлені на головну сторінку застосунку, де зможете переглядати доступні приміщення. Якщо ви забули пароль, не хвилюйтеся. На сторінці входу є посилання «Забули пароль?». Натиснувши на нього, ви перейдете до процесу відновлення пароля, де вам буде запропоновано ввести вашу електронну пошту. Після цього ви отримаєте інструкції на свою електронну пошту для скидання пароля.

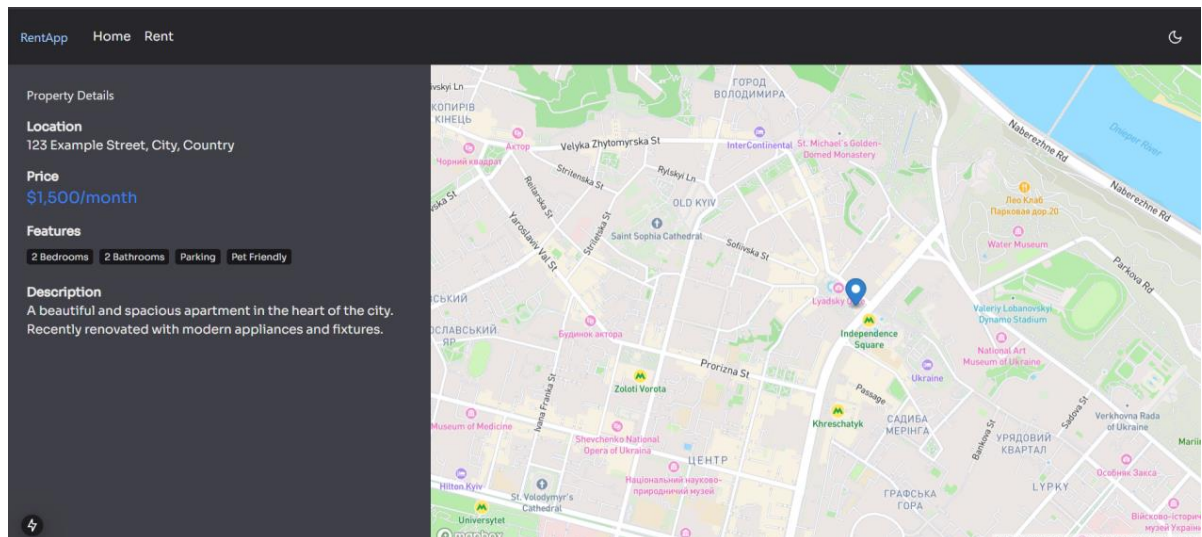


Рисунок 4.2 - Перегляд на мапі

Після успішного входу в систему ви можете почати шукати приміщення для оренди за допомогою інтерактивної карти. Для цього знайдіть вкладку або кнопку «Карта» в меню навігації на головній сторінці та натисніть на неї. На екрані з'явиться інтерактивна карта, яка відображає різні доступні приміщення для оренди. Використовуйте функцію масштабування на карті, щоб збільшити або зменшити масштаб і детальніше розглянути конкретні райони (рис. 4.2).

На карті будуть позначені маркери, які представляють доступні приміщення. Клацнувши на маркер, ви зможете переглянути деталі конкретного приміщення, такі як фотографії, опис і ціну.

Якщо ви хочете звузити результати пошуку, скористайтесь панеллю фільтрів (якщо така доступна). Ви можете налаштувати фільтри за різними параметрами, такими як ціна, тип приміщення або кількість кімнат. Після застосування фільтрів карта оновиться відповідно до ваших критеріїв, що дозволить вам швидше знайти потрібний об'єкт. Коли ви натискаєте на маркер, з'явиться спливаюче вікно з додатковою інформацією про приміщення. Тут ви знайдете контактну інформацію власника, переглянете галерею фотографій та дізнаєтеся про умови оренди. Якщо вас зацікавило конкретне приміщення, ви можете натиснути кнопку «Зберегти» або

«Забронювати», щоб перейти до процесу бронювання. У процесі бронювання вам буде запропоновано заповнити форму з вашими даними та умовами оренди. Дотримуйтесь інструкцій на екрані для завершення процесу бронювання.

Ця інструкція допоможе вам швидко зорієнтуватися у веб-застосунку для оренди приміщень та ефективно використовувати його функціонал для знаходження і бронювання бажаних об'єктів. Якщо у вас виникнуть додаткові питання або проблеми, звертайтеся до служби підтримки через відповідний розділ у застосунку.

Інші можливості веб-застосунку для оренди приміщень:

- зміна налаштувань профілю: Користувачі можуть редагувати свої особисті дані, такі як ім'я, електронна пошта, номер телефону та пароль, щоб підтримувати актуальність інформації;
- перегляд історії бронювань: Користувачі мають можливість переглядати всі свої минулі та активні бронювання, а також статус кожного з них, що дозволяє легко відстежувати орендні угоди;
- отримання сповіщень: Користувачі можуть налаштовувати отримання сповіщень про нові пропозиції, зміни в статусі бронювання або відповіді на їхні запити, що допомагає залишатися в курсі важливих оновлень;
- система оцінок і відгуків: Користувачі можуть залишати відгуки про приміщення та оцінювати їх за різними критеріями, що сприяє формуванню довіри та допомагає іншим користувачам у виборі.

4.3 Тестування веб-застосунку

У цьому прикладі ми розглянемо, як виконати функціональне тестування компонента в Next.js за допомогою React Testing Library [29, с. 510]. Ми створимо простий компонент, який відображає список приміщень, і напишемо тести для перевірки його роботи.

Спочатку створимо новий проект Next.js, якщо у вас його ще немає. Використайте команду `npx create-next-app my-rental-app`, а потім перейдіть у каталог проекту за допомогою `cd my-rental-app`. Далі встановіть необхідні пакети для тестування, виконавши команду `npm install --save-dev jest @testing-library/react @testing-library/jest-dom`.

Тепер створимо компонент, який відображатиме список приміщень. Додайте файл `RentalList.tsx` у каталог `components` з таким кодом:

```
```tsx
// components/RentalList.tsx
import React from 'react';

interface Rental {
 id: number;
 name: string;
}
interface RentalListProps {
 rentals: Rental[];
}
const RentalList: React.FC<RentalListProps> = ({ rentals }) => {
 return (
 <div>
 <h2>Список приміщень</h2>

 {rentals.map((rental) => (
 <li key={rental.id}>{rental.name}
))}

 </div>
);
};
export default RentalList;
```
```

Цей компонент приймає масив об'єктів `rentals`, кожен з яких має `id` і `name`, і відображає їх у списку.

Тепер напишемо тести для цього компонента, щоб перевірити, чи він працює правильно. Створіть файл `RentalList.test.tsx` у каталозі `__tests__` з таким кодом:

```

``tsx
// __tests__/RentalList.test.tsx
import React from 'react';
import { render, screen } from '@testing-library/react';
import RentalList from '../components/RentalList';
describe('RentalList', () => {
  test('renders the rental list correctly', () => {
    const rentals = [
      { id: 1, name: 'Квартира в центрі' },
      { id: 2, name: 'Будинок на околиці' },
    ];
    render(<RentalList rentals={rentals} />);
    const heading = screen.getByText(/Список приміщень/i);
    expect(heading).toBeInTheDocument();
    const firstRental = screen.getByText(/Квартира в центрі/i);
    const secondRental = screen.getByText(/Будинок на околиці/i);
    expect(firstRental).toBeInTheDocument();
    expect(secondRental).toBeInTheDocument();
  });
});

```

У цьому тесті ми використовуємо функцію `render` для рендерингу компонента `RentalList` з масивом приміщень. Потім ми перевіряємо, чи заголовок «Список приміщень» відображається на екрані, а також чи присутні елементи списку для кожного приміщення.

Після написання тестів додайте скрипт для запуску тестів у ваш файл `package.json`, щоб ви могли легко їх виконувати. Додайте `«test»: »jest»` у секцію «scripts». Тепер ви можете запустити тести командою `npm test`.

Цей приклад демонструє функціональне тестування компонента в Next.js за допомогою React Testing Library. Завдяки цьому ви можете перевірити, чи працює ваш компонент так, як очікується, що є важливим кроком у забезпеченні якості вашого веб-застосунку.

Навантажувальне тестування веб-застосунку є важливим етапом, який дозволяє оцінити, як система справляється з очікуваним навантаженням. Для цього можна використовувати Apache JMeter, потужний інструмент для тестування продуктивності. Далі наведено приклад, як налаштувати навантажувальний тест за допомогою JMeter [5].

Спочатку вам потрібно завантажити та встановити JMeter. Після розпакування файлів програми запустіть JMeter, відкривши файл `jmeter.bat` (для Windows) або `jmeter` (для Linux/Mac) у каталозі `bin`.

Після запуску JMeter створіть новий план тестування. Для цього натисніть правою кнопкою миші на «Test Plan» і виберіть «Add» -> «Threads (Users)» -> «Thread Group». У групі потоків ви можете налаштувати параметри:

- Number of Threads (users): Вкажіть кількість користувачів, які ви хочете емулювати. Наприклад, 1000 користувачів;
- Ramp-Up Period (in seconds): Визначте час, за який всі користувачі повинні почати виконувати запити. Наприклад, якщо ви встановите 100 секунд, то JMeter буде запускати по 10 користувачів на секунду;
- Loop Count: Вкажіть, скільки разів кожен користувач повинен виконати запит. Наприклад, 5 разів.

Далі додайте HTTP-запит до вашого плану тестування. Для цього натисніть правою кнопкою миші на «Thread Group» і виберіть «Add» -> «Sampler» -> «HTTP Request». У вікні налаштувань заповніть поля:

- Name: Введіть назву для запиту;
- Server Name or IP: Введіть домен або IP-адресу сервера (наприклад, `example.com`);

- Path: Введіть шлях до ресурсу, який ви хочете протестувати (наприклад, `/api/v1/rentals`);
- Method: Виберіть метод запиту (GET або POST).

Після налаштування HTTP-запиту ви можете додати плагін для відображення результатів тестування. Клацніть правою кнопкою миші на «Thread Group», виберіть «Add» -> «Listener» -> «View Results in Table». Це дозволить вам бачити результати тестування в реальному часі.

Коли все налаштовано, збережіть ваш тестовий план через меню «File» -> «Save Test Plan as». Тепер ви готові до запуску тесту. Натисніть зелену кнопку «Start» у верхньому меню.

Після завершення тестування ви зможете проаналізувати результати. У вкладці «View Results in Table» ви побачите дані про час відгуку, кількість успішних та неуспішних запитів та інші важливі метрики.

Цей приклад демонструє базовий процес налаштування навантажувального тестування за допомогою Apache JMeter. Використання JMeter дозволяє ефективно перевіряти продуктивність веб-застосунків під різними умовами навантаження і виявляти потенційні проблеми ще до їх виникнення в реальних умовах експлуатації.

4.4 Розгортання веб-застосунку

Наведемо детальну інструкцію для розгортання вже існуючого веб-застосунку, який базується на Next.js та інтегровано з Supabase. Інструкція розрахована на тих, хто має готовий проект, збережений у репозиторії, і бажає розгорнути його локально для розробки, а згодом – у продакшн-оточенні.

Першим кроком є клонування репозиторію з існуючим проектом. Для цього відкрийте термінал та виконайте команду, що відповідає вашій системі контролю версій, наприклад,

```
git clone https://github.com/to/be/exposed
```

Після завершення клонування перейдіть у каталог проекту:

```
cd exposed-later
```

Цей крок забезпечує наявність актуальної версії коду на вашій локальній машині. Після того, як ви отримали копію проекту, необхідно встановити всі залежності. Для цього виконайте команду:

```
npm install
```

Ця команда зчитає список пакетів із файлу `package.json` і завантажить усі необхідні модулі, що є основою для подальшої роботи застосунку.

Якщо проект використовує змінні оточення для підключення до Supabase або інших сервісів, переконайтеся, що файл конфігурації (`.env`, `.env.local` чи подібний) знаходиться у корені проекту. Якщо його немає, створіть файл (наприклад, `.env.local`) і додайте до нього наступні параметри, заповнивши їх відповідними значеннями:

```
NEXT_PUBLIC_SUPABASE_URL=https://your-project-ref.supabase.co
```

```
NEXT_PUBLIC_SUPABASE_ANON_KEY=your-anon-key
```

Використання префіксу `NEXT_PUBLIC_` гарантує, що ці змінні будуть доступні як на стороні клієнта, так і на сервері, що є критично важливим для коректної роботи застосунку з API Supabase.

Після налаштування змінних оточення, якщо у вашому проекті використовуються міграційні файли для управління структурою бази даних, переконайтеся, що у вас встановлений Supabase CLI. Якщо він ще не встановлений, ви можете встановити його глобально або використовувати через `prx`. Для запуску міграційних файлів виконайте наступну команду:

```
prx supabase migration run
```

Ця команда прочитає всі невиконані міграції та застосує їх до вашої бази даних, що дозволить синхронізувати структуру таблиць із останніми змінами у проекті. Після завершення всіх підготовчих кроків, наступним етапом є запуск локального сервера розробки. Для цього використовуйте команду:

```
npm run dev
```

Запуск сервера дозволяє перевірити роботу застосунку у режимі розробки, переглянути сторінки у браузері за локальною адресою (зазвичай `http://localhost:3000`) та внести необхідні корективи у випадку виявлення помилок або неточностей.

Коли ви переконаєтеся, що застосунок працює належним чином у локальному середовищі, можна переходити до розгортання проекту у продакшн-оточенні. Якщо ви плануєте розмістити застосунок на платформах, таких як Vercel, зверніть увагу на наступне: переконайтеся, що всі необхідні змінні оточення налаштовано у налаштуваннях платформи, а також вкажіть команду збірки та інші специфічні параметри, необхідні для правильної роботи проекту.

Наприклад, у випадку розгортання на Vercel необхідно створити новий проект, підключити репозиторій з вашим застосунком, встановити змінні оточення (`NEXT_PUBLIC_SUPABASE_URL`, `NEXT_PUBLIC_SUPABASE_ANON_KEY` тощо) та виконати автоматичну збірку проекту. Після завершення процесу розгортання платформа надасть URL-адресу, за якою буде доступний ваш застосунок.

Таким чином, розгортання вже існуючого проекту з використанням Next.js та Supabase включає наступні етапи: клонування репозиторію, встановлення залежностей за допомогою `npm install`, налаштування файлу конфігурації з відповідними змінними оточення, запуск міграційних файлів для оновлення бази даних, перевірку роботи застосунку через запуск локального сервера та остаточне розгортання у продакшн-оточенні. Дотримуючись цих кроків, ви забезпечите стабільну та безпечну роботу вашого веб-застосунку як у режимі розробки, так і після публікації.

ВИСНОВКИ

Отже, важливими напрямками застосування ІКТ у всьому їх різноманітті є ринок нерухомості та туристичний бізнес, які в сучасних умовах знаходяться в динамічному розвитку, не завжди позитивному. Ці два напрямки мають принаймні одну точку перетину – оренда або бронювання житлової нерухомості.

У ході виконання дипломної роботи було розроблено веб-застосунок для оренди приміщень з використанням технологій Next.js та Supabase, що відповідає сучасним вимогам до функціональності, зручності користування та безпеки. Проведений аналіз існуючих рішень на ринку нерухомості дозволив сформулювати вимоги до системи, які враховують як потреби орендодавців, так і очікування потенційних орендарів. Завдяки використанню фреймворка Next.js забезпечено високопродуктивну та масштабовану архітектуру застосунку, а інтеграція з Supabase дозволила реалізувати надійну систему управління даними та автентифікації користувачів.

Під час розробки були детально опрацьовані концептуальна модель системи, технічне завдання та алгоритми взаємодії між компонентами застосунку, що сприяло формуванню єдиної та цілісної архітектури рішення. Реалізація веб-застосунку підтвердила доцільність використання сучасних технологій для автоматизації процесів оренди приміщень, що дозволяє зменшити час пошуку та укладення договорів, а також покращити комунікацію між усіма учасниками процесу.

Отримані результати демонструють можливість подальшого розвитку та масштабування системи з урахуванням змін ринку нерухомості та потреб користувачів. Розроблений застосунок має високий потенціал для впровадження як у великих компаніях, так і серед малих підприємств, що займаються орендою приміщень. Практичне значення роботи полягає у створенні ефективного інструменту, який сприятиме підвищенню

конкурентоспроможності підприємств та оптимізації операційних процесів в галузі нерухомості.

Таким чином, мета дипломної роботи, що полягала у розробці веб-застосунку для оренди приміщень та експериментальній перевірці його ефективності, була досягнута, а всі поставлені у ході виконання дипломної роботи задачі виконані.

СПИСОК ЛІТЕРАТУРИ

1. Білозубенко В. С., Поляков М. П., Шаблій С. Є. Перспективи розвитку глобальної готельної індустрії. Економічний простір. 2021. № 165. С. 18–20.
2. Все про нерухомість: топ 7 безкоштовних онлайн-сервісів. URL: <https://kosht.media/top-7-bezkoshtovnykh-onlayn-servisiv-pro-nerukhomist/>.
3. Вивюрка А., Мариненко Л., Нога О., Хоміцький Б., Ланевич Т. Дослідження ефективності процесів CI/CD в гнучких технологіях розробки програмного забезпечення. Матеріали XII Міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій». 2023. С. 402–403.
4. Жеребцов О., Іжиков А. Використання функціонального програмування у JavaScript та фреймворках. Інноваційні рішення в інженерії програмного забезпечення. 2022. № 9. С. 162–169.
5. Застосування JMeter в тестуванні. URL: <https://training.qatestlab.com/blog/technical-articles/using-jmeter-in-testing/>.
6. Зінов'єва О. Г. Використання CASE-засобів для проектування інформаційних систем. Українські студії в європейському контексті. 2023. № 7. С. 220–227.
7. Ізмайлова О., Красовська Г., Пальчик П. База моделей інформаційної технології оцінки вартості житлової нерухомості. *Pidvodni tehnologii*. 2023. № 13. С. 3–13.
8. Інструментальні засоби для веб-розробки URL: <http://comscienceatschool.blogspot.com>.
9. Машіка Г. В., Косовець А. І. Розвиток туристичної галузі України в сучасних умовах. Матеріали Всеукраїнської науково-практичної інтернет-конференції «Вітчизняна наука на зламі епох: проблеми та перспективи розвитку» : зб. наук. пр. Переяслав, 2023. Вип. 91 . С. 5–7.

10. Методи розробки сайтів URL: <https://webstudio2u.net/ua/webdesign/354-sitedevelop-methods.html>.
11. Назар Б. Електронний уряд в Україні: приклади успішних проектів та їхній вплив на суспільство. Наука і техніка сьогодні. 2024. № 8(36). С. 892–892.
12. Онищук Н. В. Інноваційні напрямки у сфері туризму. Молодий вчений. 2017. № 8. С. 456–459.
13. Садовяк М. С. Аналіз факторів ринку оренди житлової нерухомості в Україні. Збірник наукових праць ЧДТУ. Серія: Економічні науки. № 52. С. 21–29.
14. Тестування навантаження сайту або сервера: інструменти. URL: <https://highload.tech/uk/blogs/yak-provesti-testuvannya-navantazhennya-shho-potribno-znati-ta-yakimi-instrumentami-koristuvatisya/>.
15. Павлов В., Кривов'язюк І. Механізми стимулювання розвитку сучасного ринку нерухомості. Регіональна економіка. 2005. № 2. С. 67–75.
16. Переваги використання Next.js у сучасній розробці веб-додатків. URL: <http://ir.nmu.org.ua/handle/123456789/164328>.
17. Фесак С. А. Ринок нерухомості як об'єкт державного регулювання в Україні. Публічне управління: теорія та практика. 2013. № 1. С. 147–154.
18. Цвілій С.М., Зайцева В.М., Сокол К.М. Повоєнний розвиток підприємств індустрії туризму на основі впровадження інформаційних технологій. Українські студії в європейському контексті. 2023. №7. С. 381–388.
19. Чуєва І., Жестков С., Сидорук А. Сучасні тенденції розвитку онлайн бронювання туристичних послуг в Україні. Економіка та суспільство. 2021. № 27. С. 1–4.
20. Шаповал В. М. Розробка веб-додатку для бронювання та обміну житлом. Міжнародна науково-практична конференція молодих вчених, аспірантів та студентів «Інформаційні технології в сучасному світі: дослідження молодих вчених»: тези доповідей (12-13 березня 2015 р., м. Харків). 2015. С. 73.

21. Шаров С. В., Лубко Д. В., Зинов'єва О. Г. Використання інтелектуальних систем у туристичному бізнесі. Таврійський науковий вісник. Серія: Технічні науки. 2022. № 1. С. 69–75.
22. Шаров С. В., Скрипка С. О. Використання мови UML для інфологічного моделювання реляційної бази даних. Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення. 2014. № 7. С. 5–9.
23. Шаров С. В., Шамардак О. А. Концептуальне моделювання інформаційної системи за допомогою мови UML. Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення. 2014. № 7. С. 10–14.
24. Шевчук М. І. Веб-застосунок для автоматизації пошуку об'єктів нерухомості : кваліфікаційна робота бакалавра : 121 Інженерія програмного забезпечення. Хмельниц. нац. ун-т. Хмельницький, 2023. 118 с.
25. Що таке Next js і для чого потрібен. URL: <https://pxstudio.pw/blog/chto-takoe-next-js-i-dlya-chego-on-nuzhen>.
26. Яцук М.С. WEB-сайт сервісу для бронювання люкс апартаментів та п'ятизіркових готелів : дипломна робота на здобуття ступеня бакалавра спеціальності «Комп'ютерні науки», «Інформаційні управляючі системи та технології». Київ, 2023. 64 с.
27. Ballamudi V. K. R. et al. Getting Started Modern Web Development with Next. js: An Indispensable React Framework. Digitalization & Sustainability Review. 2021. Т. 1. № 1. С. 1–11.
28. Booking.com. Найкращі готелі, авіаквитки, авто напрокат і помешкання. URL: <https://www.booking.com>.
29. GriffithsD., Griffiths D. React Cookbook: Recipes for Mastering the React Framework. 2021. 510 p.
30. Getting Started: Project Structure. Next.js. URL: <https://nextjs.org/docs/app/getting-started/project-structure>.

31. Ichanska N., Gritsenko A., Shefer V. Development of software service for sale of vehicles. *Системи управління, навігації та зв'язку*. 2018. Т 4. № 50. С. 105–108.
32. Kraynyuk L. M. et al. Digital innovations in the tourism and hospitality sector of the Kharkiv region. *Бизнес информ*. 2020. № 7. С. 100–112.
33. Next.js With TypeScript Development, URL: <https://www.wearecapicua.com/blog/nextjs-with-typescript>.
34. PostgreSQL Primary Key. URL: <https://neon.tech/postgresql/postgresql-tutorial/postgresql-primary-key>.
35. Setting up Server-Side Auth for Next.js. *Supabase Docs*. URL: <https://supabase.com/docs/guides/auth/server-side/nextjs>.
36. RoomGuru.com. Глобальний пошуковик, який порівнює мільйони дешевих авіарейсів, готелів та сервісів оренди авто. URL: <https://www.momondo.ua>.
37. Hotels.com. The company offers a one-stop shopping source for hotel pricing, amenities and availability. URL: <https://www.hotels.com>.
38. Lun.ua. Онлайн-платформа для пошуку нерухомості в Україні. URL: <https://www.lun.ua>.
39. DIM.RIA™ – вся нерухомість України. Продаж і оренда будь-якої нерухомості. URL: <https://dom.ria.com/uk/>.
40. Why I chose Supabase as BaaS for my projects. *Indie Hackers*. URL: <https://www.indiehackers.com/post/why-i-chose-supabase-as-baas-for-my-projects-aa666da293>.
41. Top 10 tools for faster development in Next.js. URL: <https://dev.to/hamzakhan/top-10-tools-for-faster-development-in-nextjs-465i>.
42. Version Control Systems and Their Importance - DEV Community .URL: <https://dev.to/shariqahmed525/version-control-systems-and-their-importance>.
43. Zillow: Real Estate, Apartments, Mortgages & Home Values. URL: <https://www.zillow.com/>.

ДОДАТОК А

Код (лістинг) програми

app/page.tsx

```

import {
  Box,
  Container,
  Heading,
  Text,
  VStack,
  Button,
} from »@chakra-ui/react»;
import Link from »next/link»;

const MainPage = () => {
  return (
    <Box
      pt=»72px»
      minH=»100vh»
    >
      <Container maxW=»container.xl» py={10}>
        <VStack gap={8} align=»stretch»>
          <Box textAlign=»center»>
            <Heading size=»2xl» mb={4}>
              Welcome to Rent App
            </Heading>
            <Text
              fontSize=»xl»
            >
              Your one-stop solution for property management
            </Text>
          </Box>

          <Box
            p={8}
            borderRadius=»xl»
            boxShadow=»xl»
          >
            <VStack gap={6}>
              <Link href=»/browse»>
                <Button
                  colorScheme=»blue»
                  size=»lg»
                  w=»full»
                >
                  Browse Properties
                </Button>
              </Link>
              <Button colorScheme=»green» size=»lg» w=»full»>
                List Your Property
              </Button>
            </VStack>
          </Box>
        </VStack>
      </Container>
    </Box>
  )
}

```

```

        </VStack>
      </Box>
    </VStack>
  </Container>
</Box>
);
};
export default MainPage;

```

app/layout.tsx

```

import type {Metadata} from »next»;
import {Provider} from »@/components/ui/provider»;
import {ReactNode} from »react»;
import »./globals.css»;
import {Box, Flex} from »@chakra-ui/react»;
import Navbar from »@/components/Navbar»;
import {sora} from »@/utils/fonts»;

export const metadata: Metadata = {
  title: »Next App«,
  description: »Next App«,};

export default function RootLayout({children}: Readonly<{
  children: ReactNode;
}>) {
  return (
    <html lang=»en» suppressHydrationWarning>
      <body className={sora}>
        <Provider>
          <Flex
            overflow=»hidden»
            direction=»column»
            maxH=»dvh»
            h=»dvh»
          >
            <Navbar/>
            <Box h=»full»>{children}</Box>
          </Flex>
        </Provider>
      </body>
    </html> );}

```

app/auth/components/AuthForm.tsx

```

'use client';
import {useState} from 'react';
import {
  Box,
  Flex,
  Input,
  VStack,

```

```

Heading,
Text,
} from '@chakra-ui/react';
import {Button} from »@/components/ui/button»;
import {Mail, Lock} from 'lucide-react';
import {useColorModeValue} from »@/components/ui/color-mode»;
import {login, signup} from '../actions';
import {useForm} from »react-hook-form»;
import {PasswordInput} from »@/components/ui/password-input»;
import {Field} from »@/components/ui/field»;
type AuthMode = 'login' | 'signup';
interface FormInputs {
  email: string;
  password: string;}
export function AuthForm() {
  const [mode, setMode] = useState<AuthMode>('login');
  const {
    register,
    handleSubmit,
    formState: {errors, isSubmitting},
    reset
  } = useForm<FormInputs>();

  const onSubmit = async (data: FormInputs) => {
    try {
      if (mode === 'login') {
        await login(data);
      } else {
        await signup(data);
      }
    } catch (error) {
      console.error(error);
    }
  };

  const bgColor = useColorModeValue('gray.100', 'gray.500');
  const boxBg = useColorModeValue('white', 'gray.600');
  return (
    <Flex
      width=»full»
      height=»full»
      align=»center»
      justifyContent=»center»
      bg={bgColor}
    >
      <Box
        borderWidth={1}
        px={4}
        width=»full»
        maxWidth=»500px»
        borderRadius={4}
        textAlign=»center»
        boxShadow=»lg»
        bg={boxBg}

```

```

    p={8}
  >
  <Heading mb={6}>
    {mode === 'login' ? 'Sign In' : 'Create Account'}
  </Heading>

  <form onSubmit={handleSubmit(onSubmit)}>
    <VStack>

      <Field label=»Email« invalid={!!errors.email} errorText={errors.email?.message}>
        <Flex w=»full«
          align=»center«
          justifyContent=»center«
          gap={3}>
          <Mail/>
          <Input
            className=»bg-white text-black«
            {...register('email', {
              required: 'Email is required',
              pattern: {
                value: /\S+@\S+\.\S+/,
                message: 'Invalid email address'
              }
            })}
          />
        </Flex>
      </Field>

      <Field label=»Password« invalid={!!errors.password}
errorText={errors.password?.message}>
        <Flex w=»full«
          align=»center«
          justifyContent=»center«
          gap={3}>
          <Lock/>
          <PasswordInput
            className=»bg-white text-black«
            type=»password«
            {...register('password', {
              required: 'Password is required',
              minLength: {
                value: 6,
                message: 'Password must be at least 6 characters'
              }
            })}
          />
        </Flex>
      </Field>
      <Button
        type=»submit«
        width=»full«
        loading={isSubmitting}
      >
        {mode === 'login' ? 'Sign In' : 'Sign Up'}
      </Button>

```

```

<Text textAlign=»center»>
  {mode === 'login' ? »Don't have an account? » : »Already have an account? »}
  <Button
    variant=»ghost»
    colorScheme=»blue»
    onClick={() => {
      setMode(mode === 'login' ? 'signup' : 'login');
      reset();
    }}
  >
    {mode === 'login' ? 'Sign up' : 'Sign in'}
  </Button>
</Text>
</VStack>
</form>

</Box>
</Flex> );}

```

app/browse/page.tsx

```

»use client»;
import {Else, If, Then} from »react-if»;
import {Box, Text, VStack, Heading, Badge, Flex, Skeleton} from »@chakra-ui/react»;
import {useColorMode} from »@/components/ui/color-mode»;
import {Coordinate} from »@/lib/types»;
import MapBox from »@/components/MapBox»;
import {Spinner} from »@/components/ui/spinner»;
const defaultPosition: Coordinate = {
  latitude: 50.4504,
  longitude: 30.5238,
};
const RentPage = () => {
  const {colorMode} = useColorMode();
  const position: Coordinate | null = defaultPosition;
  return (
    <If condition={!position}>
      <Then>
        <Flex h=»full» bg={colorMode === »light» ? »gray.50» : »gray.800»}>
          {/* Left side - Property Details */}
          <Box
            flex={1}
            overflowY=»auto»
            bg={colorMode === »light» ? »white» : »gray.700»}
            p={6}
          >
            <VStack align=»stretch» gap={4}>
              <Heading size=»lg»>Property Details</Heading>
              <Box>
                <Text fontWeight=»bold»>Location</Text>
                <Text>123 Example Street, City, Country</Text>
              </Box>
            </VStack>
          </Box>
        </Then>
      </If>
    )
  );
}

```

```

<Box>
  <Text fontWeight=»bold»>Price</Text>
  <Text fontSize=»xl» color=»blue.500»>
    $1,500/month
  </Text>
</Box>
<Box>
  <Text fontWeight=»bold»>Features</Text>
  <Box mt={2}>
    <Badge mr={2} mb={2}>
      2 Bedrooms
    </Badge>
    <Badge mr={2} mb={2}>
      2 Bathrooms
    </Badge>
    <Badge mr={2} mb={2}>
      Parking
    </Badge>
    <Badge mr={2} mb={2}>
      Pet Friendly
    </Badge>
  </Box>
  <Box>
    <Text fontWeight=»bold»>Description</Text>
    <Text>
      A beautiful and spacious apartment in the heart of the city.
      Recently renovated with modern appliances and fixtures.
    </Text>
  </Box>
</VStack>
</Box>
{ /* Right side - Map Box */}
<Box flex={2}>
  <Skeleton
    backgroundColor={colorMode === »light» ? »gray.200» : »gray.600»}
    h=»full»
    w=»full»
    loading={!position}
  >
    {position && (
      <MapBox
        latitude={position.latitude}
        longitude={position.longitude}
      />
    )}
  </Skeleton>
</Box>
</Flex>
</Then>
<Else>
  <Spinner global size=»xl»/>
</Else>
</If>
);};
export default RentPage;

```

ДОДАТОК Б

Стаття

*Матеріали IV Всеукраїнської науково-практичної Інтернет-конференції
«Сучасні комп'ютерні та інформаційні системи і технології»*

Подальші дослідження спрямовані на розробку універсальних рішень для автоматизації, адаптованих до різних закладів освіти.

Список використаних джерел:

1. Moodle Documentation [Електронний ресурс]. – Режим доступу: <https://docs.moodle.org>.
2. GitHub – офіційна сторінка сервісу [Електронний ресурс]. – Режим доступу: <https://github.com/>.
3. GNU Bash, Awk, sed, tar, gzip, grep, wget – офіційна сторінка [Електронний ресурс]. – Режим доступу: <https://www.gnu.org/software/>.
4. ShellCheck – офіційна сторінка інструмента [Електронний ресурс]. – Режим доступу: <https://www.shellcheck.net/>.
5. Rsync – офіційна сторінка [Електронний ресурс]. – Режим доступу: <https://rsync.samba.org/>.
6. Curl – офіційна сторінка [Електронний ресурс]. – Режим доступу: <https://curl.se/>.
7. Сокорчук І. П. Автоматизація повсякденних завдань викладачів за допомогою засобів штучного інтелекту: покращення робочого процесу та ефективності / І. П. Сокорчук // Поліграфічні, мультимедійні та web-технології : тези доп. IX Міжнар. наук.-техн. конф., 14–18 травня 2024 р. – Т. 1. – Харків: ТОВ «Друкарня Мадрид», 2024. – С. 330–331.

УДК 004.738.5

**ВИКОРИСТАННЯ ІКТ У ГОТЕЛЬНОМУ БІЗНЕСІ ТА РИНКУ
НЕРУХОМОСТІ**

Тарасюк В.Р., здобувач вищої освіти *e-mail: djwar0962@gmail.com*
Науковий керівник: к.пед.н., доцент Шаров С.В.
*Таврійський державний агротехнологічний університет імені Дмитра
Моторного*

Актуальність та постановка проблеми. В умовах інформатизації суспільства та впровадження інформаційно-комунікаційних технологій (ІКТ) у різні напрямки діяльності та виробництва широке розповсюдження отримали різноманітні онлайн сервіси. Їх використання дозволяє залучити більшу цільову аудиторію, покращити маркетингову діяльність, здійснювати онлайн комерцію тощо. Важливими напрямками застосування ІКТ є підтримка діяльності туристичного бізнесу, а також ринку нерухомості. В межах цих напрямків клієнти можуть здійснювати бронювання та аренду житлових та нежитлових приміщень, відслідковувати статус замовлень, комунікувати з орендодавцями та туристичними операторами тощо.

Зважаючи на важливість забезпечення цифровізації економічної діяльності, метою дослідження є аналіз переваг та можливостей ІКТ в туристичному бізнесі та функціонуванні ринку нерухомості. Отримані знання допоможуть визначити найбільш перспективні напрямки впровадження онлайн сервісів в даних сферах.

Основні матеріали дослідження. В умовах жорсткої конкуренції підприємства запроваджують нові форми взаємодії з клієнтами, сучасні засоби

реалізації товарів та послуг через мережу Інтернет. І це є об'єктивним кроком на шляху підвищення власної конкурентоспроможності, адже кількість Інтернет-ресурсів зростає з кожним днем. Добре продуманий та просунутий веб ресурс дозволяє отримати більший прибуток, ніж робота декількох операторів (співробітників підприємства), які самотужки шукають клієнтів та партнерів, здійснюють комунікацію тощо. Крім того, у разі використання Інтернет-ресурсів відсутні витрати на оренду дороги, проживання під час відрядження, інші витрати [9, с. 105].

Важливими напрямками застосування ІКТ у всьому їх різноманітті є ринок нерухомості та туристичний бізнес, які в сучасних умовах знаходяться в динамічному розвитку, не завжди позитивному. Ці два напрямки мають принаймні одну точку перетину – оренда або бронювання житлової нерухомості. І від того, наскільки якісно буде забезпечено інформування клієнтів про житло, умови бронювання/оренду, способи оплати тощо, буде залежати успішність оформлених угод.

Якщо розглянути можливості ІКТ для оптимізації функціонування ринку нерухомості та підвищення його ефективності, то можна виокремити декілька переваг. У випадку житлових приміщень потенційні клієнти можуть переглянути деталі житла, здійснити фільтрацію знайдених варіантів, прочитати відгуки тощо. Якщо потрібно знайти нежитлове приміщення (наприклад, офісне або складське приміщення), то за допомогою параметрів пошуку можна швидко знайти приміщення відповідно до потреб компанії.

Оскільки для онлайн сервісів практично немає обмежень у поширенні інформації про житловий та нежитловий фонд серед цільової аудиторії, це дозволяє орендодавцям вийти на регіональний або навіть міжнародний ринок. Потенційні клієнти можуть орендувати житло або офіси в інших країнах, використовуючи міжнародні платформи Airbnb, Booking.com, Zillow та ін. Це надає можливість збільшити діапазон варіантів житла та підібрати квартиру за оптимальну ціну. У роботі [8] звертається увага на таких платформах як ЛУН, Dom Ria, Zillow.com, які широко використовуються в українському сегменті. Онлайн платформа ЛУН орієнтована виключно на житлову нерухомість, позиціонує себе як найбільший український сайт для пошуку квартир. Сервіс Dom Ria зосереджений на житловій нерухомості, накопичує інформацію після її введення безпосередньо користувачами. Провідний міжнародний сервіс Zillow призначений для пошуку житлової нерухомості та охоплює всі аспекти життєвого циклу володіння нерухомістю.

Через використання спеціалізованих онлайн платформ можна досягти певного рівня прозорості та довіри до орендодавців. Такі сервіси часто включають рейтингові системи серед орендодавців і орендарів, що дозволяє уникати недобросовісних угод. Крім того, зазвичай такі платформи містять достатньо детальну інформацію про житлові та нежитлові приміщення, зокрема дані про попередніх орендарів, технічний стан приміщення, юридичні аспекти угоди тощо. Прикладом може слугувати система електронних закупівель «Prozorro», що була розроблена з метою підвищення прозорості та підзвітності державних закупівель. Онлайн платформа дозволяє здійснювати всі етапи закупівельного процесу в електронному вигляді, забезпечуючи рівні права для всіх учасників торгів [3, 888].

Наступним напрямком економічної діяльності, де можна застосувати інформаційно-комунікаційні технології для оренди житлової нерухомості, є туристичний бізнес. Науковці зазначають, що готельний та ресторанний бізнес є

однією з галузей, яка найбільш динамічно розвивається в напрямі застосування ІКТ. Актуальним є впровадження інновацій для створення нових конкурентоспроможних туристичних продуктів, формування сприятливих умов для залучення інвестицій та активізації туристичної діяльності [4, с. 456]. Як наслідок, інформаційні технології у сфері туризму дозволяють користувачу самостійно планувати подорож, спілкуватися з місцевими жителями за допомогою синхронного перекладу через мобільний перекладач, отримувати персоналізовані рекомендації та ін. Крім того, впровадження ІКТ у туристичну індустрію є закономірним кроком на шляху цифровізації туристичної діяльності, що відповідає світовим стандартам розвитку передових технологій [1, с. 21].

Туристичні компанії для підвищення конкурентоспроможності серед інших туроператорів активно застосовують інформаційні технології для досягнення двох основних цілей: маркетингу та прямих продажів своїх послуг [5, с. 386]. З метою підвищення ефективності обслуговування клієнтів і розширення асортименту послуг вони використовують спеціалізоване програмне забезпечення, у тому числі засноване на використанні штучного інтелекту [7, с. 70].

З іншого боку, можна скористатися значними перевагами, що надає мережа Інтернет. Хоча традиційні джерела інформації, такі як туристичні агенти та телебачення, залишаються актуальними, все більше туристів віддають перевагу онлайн-замовленням послуг, використанню онлайн-туристичних агентств та самостійному бронюванню квитків в авіакомпаніях, готелях тощо [1, с. 20]. Перспективи розвитку індустрії гостинності тісно пов'язані із зростанням ролі онлайн-бронювань та соціальних мереж, які стають основними каналами для резервування готелів і ресторанів [2, с. 4]. Одним із відомих онлайн сервісів з бронювання житла є Booking.com. Перевага сайту полягає у можливості не оплачувати проживання заздалегідь, а здійснювати оплату безпосередньо під час заселення. Крім того, сервіс має гнучку систему знижок на бронювання житла. Такий підхід у бронювання житла забезпечив моделі Booking.com домінуючі позиції на ринку бронювання та оренди нерухомості [6].

Висновки. Отже, впровадження ІКТ у ринок нерухомості дозволяє зробити його більш зручним, прозорим і доступним як для орендодавців, так і для орендарів. Інформаційно-комунікаційні технології відкривають нові перспективи для розвитку даної сфери, полегшують доступ до об'єктів нерухомості та підвищують ефективність взаємодії між сторонами.

Впровадження та використання ІКТ у готельному бізнесі надає можливість підвищити рівень інформування, обслуговування та задоволеності клієнтів. Завдяки персоналізації послуг, автоматизації окремих операцій та аналізу даних заклади готельно-ресторанного бізнесу отримують нові можливості для залучення нових та утримування постійних клієнтів, таким чином забезпечуючи високу конкуренцію серед аналогічних компаній.

Список використаних джерел:

1. Білозубенко В.С., Поляков М.П., Шаблій С.Є. Перспективи розвитку глобальної готельної індустрії. *Економічний простір*. 2021. № 165. С. 18–22.
2. Машіка Г.В., Косовець А.І. Розвиток туристичної галузі України в сучасних умовах. *Матеріали Всеукраїнської науково-практичної інтернет-конференції «Вітчизняна наука на зламі епох: проблеми та перспективи розвитку»* : зб. наук. пр. Переяслав, 2023. Вип. 91. С. 5–7.
3. Назар Б. Електронний уряд в Україні: приклади успішних проєктів та їхній вплив на суспільство. *Наука і техніка сьогодні*. 2024. № 8(36). С. 892–892.

4. Онищук Н.В. Інноваційні напрямки у сфері туризму. *Молодий вчений*. 2017. № 8. С. 456–459.
5. Цвілий С.М., Зайцева В.М., Сокол К.М. Повоєнний розвиток підприємств індустрії туризму на основі впровадження інформаційних технологій. *Українські студії в європейському контексті*. 2023. №7. С. 381–388.
6. Чуєва І., Жестков С., Сидорук А. Сучасні тенденції розвитку онлайн бронювання туристичних послуг в Україні. *Економіка та суспільство*. 2021. № 27. С. 1–4.
7. Шаров С.В., Лубко Д.В., Зинов'єва О.Г. Використання інтелектуальних систем у туристичному бізнесі. *Таврійський науковий вісник*. Серія: Технічні науки. 2022. № 1. С. 69–75.
8. Шевчук М.І. Веб-застосунок для автоматизації пошуку об'єктів нерухомості : кваліфікаційна робота бакалавра : 121 Інженерія програмного забезпечення / М.І. Шевчук ; Хмельниц. нац. ун-т. Хмельницький, 2023. 118 с.
9. Ichanska N., Gritsenko A., Shefer V. Development of software service for sale of vehicles. *Системи управління, навігації та зв'язку*. 2018. Т 4. № 50. С. 105–108.

УДК 53 (07)

ТЕХНОЛОГІЯ МОБІЛЬНОГО НАВЧАННЯ: ПЕРЕВАГИ ТА НЕДОЛІКИ

Терещук С.І., д.п.н.

e-mail: s.i.tereschuk@gmail.com

Уманський державний педагогічний університет імені Павла Тичини

Таврійський державний агротехнологічний університет імені Дмитра Моторного

Актуальність та постановка проблеми. Донедавна головною ознакою технології мобільного навчання було використання широкого спектру цифрових та портативних мобільних пристроїв для здійснення операцій з отримання, обробки та поширення інформації [4]. Таку дефініцію можна вважати сповна застарілою. Відтепер, *мобільне навчання – це інтеграція в освітній процес цифрових технологій засобами мобільних сервісів та застосунків*. До таких сучасних технологій відноситься увесь спектр цифрових технологій від імерсивних до хмарних.

Технологія мобільного навчання набула неабиякої популярності серед освітян України [1]. Не зважаючи на критику, яка по всяк час лунає у бік освітніх систем, орієнтованих на активне використання мобільних пристроїв, дана технологія навчання активно використовується педагогами під час занять [2, 3]. Такий інтерес пов'язаний не лише з необхідністю якісної організації дистанційної освіти, але й більшою мірою новими цифровими технологіями, які стають доступнішими завдяки мобільним пристроям. Для ефективного використання мобільної технології навчання в освітньому процесі важливо з'ясувати переваги та недоліки технології мобільного навчання саме з огляду на швидкий розвиток хмарних технологій, систем штучного інтелекту, імерсивних технологій тощо.

Основні матеріали дослідження. До переваг технології мобільного навчання слід віднести:

ДОДАТОК В

Сертифікат учасника конференції

