

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Вебсайт для бронювання робочих місць з коворкінгу з використанням
рекомендаційної системи»

Ступінь вищої освіти – бакалавр
Спеціальність – 122 «Комп’ютерні науки»
Освітня програма «Комп’ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: здобувач 4 курсу
групи КН-21
Іван ОЛЕНЧУК

Керівник: викладач кафедри інформаційного
менеджменту, математики та
статистики
Олег МУШИНСЬКИЙ

Засвідчую, що кваліфікаційна
робота оформлена відповідно
до ДСТУ 3008:2015 та не
містить запозичень з праць
інших авторів без відповідних
посилань.

Здобувач: _____
(підпис)

м. Київ – 2025 рік

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»»

ЗАТВЕРДЖУЮ:
завідувач кафедри
комп'ютерних наук
_Сергій МІЧКІВСЬКИЙ
« ____ » ____ 20 ____ р

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Оленчук Іван Володимирович

Тема роботи	Вебсайт для бронювання робочих місць у коворкінгу з використанням рекомендаційної системи
Номер та дата наказу про затвердження теми	№121-7 від 24 грудня 2024 року
Коротка постановка завдання	Розробити веб-платформу для бронювання робочих місць у коворкінгу, яка забезпечує зручний вибір місця та часу, інтеграцію з платіжними сервісами та рекомендаційну систему для покращення користувацького досвіду.
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	1. Мушинський О.Ю. Особливості управління проектними командами в гібридному середовищі. <i>Економіка та суспільство</i> . 2024. № 60. DOI: 10.32782/2524-0072/2024-60-5 2. Kopplin, C. S. (2023). Software Support in Coworking Spaces: Instruments, Applications, and Challenges. <i>Awakening the Management of Coworking Spaces</i> , 97-110.
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має містити теоретичне, системотехнічне або експериментальне дослідження за темою роботи, яку слід розглядати як складне спеціалізоване завдання або практичну проблему в галузі комп'ютерних наук, яка характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.

Дата видачі завдання 27 грудня 2024 р.

Керівник

Олег МУШИНСЬКИЙ

Здобувач освітнього ступеня бакалавра

Іван ОЛЕНЧУК

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
Підготовчий етап			
1	Вибір напрямку дослідження	02.12.2024 р.	<i>виконано</i>
2	Формування теми та призначення керівника	16.12.2024 р.	<i>виконано</i>
3	Затвердження теми кваліфікаційної роботи	23.12.2024 р.	<i>виконано</i>
4	Затвердження завдання на кваліфікаційну роботу	27.12.2024 р.	<i>виконано</i>
Основний етап			
5	Розробка концепції кваліфікаційної роботи	13.01.2025 р.	<i>виконано</i>
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів	20.01.2025 р.	<i>виконано</i>
7	Затвердження розширеної постановки завдання. Підготовка та подання керівникові розділу 1 кваліфікаційної роботи	10.03.2025 р.	<i>виконано</i>
8	Проектування. Підготовка та подання керівникові розділу 2 кваліфікаційної роботи	24.03.2025 р.	<i>виконано</i>
9	Підготовка доповіді для експертизи стану виконання кваліфікаційної роботи (проміжний контроль)	31.03-04.04.2025 р.	<i>виконано</i>
10	Реалізація. Підготовка та подання керівникові розділу 3 кваліфікаційної роботи	07.04.2025 р.	<i>виконано</i>
11	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	14.04.2025 р.	<i>виконано</i>
12	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівникові доопрацьованого варіанту кваліфікаційної роботи	21.04.2025 р.	<i>виконано</i>
Завершальний етап			
13	Представлення рукопису для перевірки на плагіат	28.04-04.05.2025 р.	<i>виконано</i>
14	Підготовка презентації та доповіді на передзахист	05.05-11.05.2025 р.	<i>виконано</i>
15	Передзахист кваліфікаційної роботи	12.05-16.05.2025 р.	<i>виконано</i>
16	Доопрацювання роботи за результатами передзахисту	19.05-06.06.2025 р.	<i>виконано</i>
17	Експертиза роботи керівником та зовнішнім експертом	09.06-15.06.2025 р.	<i>виконано</i>
18	Доопрацювання доповіді та презентації для захисту	09.06-15.06.2025 р.	<i>виконано</i>
19	Захист кваліфікаційної роботи	16.06-22.06.2025 р.	<i>виконано</i>

Керівник

Олег МУШИНСЬКИЙ

Здобувач освітнього ступеня бакалавра

Іван ОЛЕНЧУК

Оленчук І.В. Вебсайт для бронювання робочих місць у коворкінгу з використанням рекомендаційної системи

Пояснювальна записка кваліфікаційної роботи за спеціальністю 122 – Комп’ютерні науки (освітня програма – Комп’ютерні науки) СО Бакалавр. – ВНЗ. «Університет економіки та права «КРОК», Навчально-науковий інститут інформаційних та комунікаційних технологій, кафедра комп’ютерних наук, Київ, 2025.

У даній роботі описано розробку веб-сайту для бронювання робочих місць у коворкінгу з використанням рекомендаційної системи

Ключові слова: веб-сайт, бронювання, робочі місця, компанії, коворкінг, рекомендації.

Рис. 18, Бібліограф. 16.

Olenchuk I.V. Website for booking jobs in coworking using a recommendation system

Explanatory note of the qualification work in the specialty 122 – Computer Science (educational program – Computer Science) SO Bachelor. – Higher education institution. “University of Economics and Law “KROK”, Educational and Scientific Institute of Information and Communication Technologies, Department of Computer Science, Kyiv, 2025.

This work describes the development of a website for booking jobs in coworking using a recommendation system

Keywords: website, booking, jobs, companies, coworking, recommendations.

Fig. 18. Bibliography: 16 Items.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ВЕБ САЙТУ	8
1.1 ПРЕДМЕТНА ОБЛАСТЬ.....	8
1.2 АНАЛІЗ ПОТЕНЦІЙНИХ КОНКУРЕНТНИХ ПЕРЕВАГ ПРОГРАМНОГО ПРОДУКТУ ..	9
1.3 ПОСТАНОВКА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	13
Висновки до розділу 1	13
РОЗДІЛ 2 ПРОЄКТУВАННЯ ВЕБ-САЙТУ ДЛЯ БРОНЮВАННЯ РОБОЧИХ МІСЦЬ У КОВОРКІНГУ	14
2.1 МОДЕЛЮВАННЯ ПОВЕДІНКИ ПРОДУКТУ.....	14
2.2 МОДЕЛЮВАННЯ СТРУКТУРИ ПРОДУКТУ	21
2.3 ОПИС АРХІТЕКТУРИ ПРОДУКТУ.....	24
Висновки до розділу 2	28
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ДЛЯ БРОНЮВАННЯ РОБОЧИХ МІСЦЬ У КОВОРКІНГУ	29
3.1 РЕАЛІЗАЦІЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ	29
3.2 ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	31
3.3 ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ	32
Висновки до розділу 3	35
ВИСНОВКИ	37
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	38
ДОДАТОК А	39

ВСТУП

Актуальність теми. Наразі сучасний світ активно переходить до цифрових рішень у різних сферах життя. Зокрема в організації робочих місць та просторів офісів. Коворкінги стали популярним вибором для фрілансерів, стартапів та будь яких компаній, які шукають зручну альтернативу традиційним звичайним офісам. І саме для ефективного використання таких просторів необхідна зручна система бронювання, яка буде ідеально підходити для будь якого вибору та швидко знаходити і резервувати робочі місця відповідно до потреб користувачів та замовників.

Таким чином, розробка веб-сайту бронювання робочих місць у коворкінгу з використанням рекомендаційної системи є достатньо актуальним та необхідним напрямком та дозволить оптимізувати розподіл робочих місць, зробивши процес реєстрації швидшим і зручним для користувачів.

Мета дослідження. Метою цього проекту є розробка веб-сайту для бронювання місць у коворкінгу, який буде використовувати рекомендаційну систему для бронювання робочих місць

Завдання на дослідження:

- ознайомитись з існуючими аналогами, виявити їхні слабкі сторони;
- розробити рекомендаційну систему бронювання робочих місць з коворкінгу, яка буде базуватися на персоналізованому виборі, а також автоматизації та підвищенню комфорту користувачів;
- спроектувати структуру веб-сайту та реалізувати прототип платформи (програмне забезпечення)

Об'єктом дослідження є персоналізований вибір, оптимізація простору, підвищення комфорту та автоматизація процесу при бронюванні робочих місць з коворкінгу

Предметом дослідження є процес бронювання робочих місць з коворкінгу та веб сайт з використанням рекомендаційної системи.

Методи дослідження: методи та алгоритми розробки сайту, аналіз існуючих рішень та даних а також рекомендаційна система на веб-сайті з бронювання робочих місць.

Практичне значення. Полягає у можливості застосування результатів роботи для вдосконалення системи бронювання коворкінгів. Розроблена платформа сприятиме на підвищення ефективності використання робочих місць, забезпеченню автоматизації процесів бронювання. Також створення веб сайту для бронювання робочих місць у коворкінку з використанням рекомендаційної системи.

Структура та обсяг пояснювальної записки. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків та списку посилань (16 найменувань) Пояснювальна записка містить 18 рисунків. Загальний обсяг пояснювальної записки складає 39 сторінок, основний зміст викладено на 35 сторінках.

РОЗДІЛ 1

ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ВЕБ САЙТУ

1.1 Предметна область

У сучасному світі коворкінг-простори [1] стали популярними серед фрілансерів, стартапів, малих компаній і навіть великих корпорацій, які прагнуть забезпечити своїм співробітникам гнучкі умови роботи та комфорт. Попит на такі простори зростає, що підвищує необхідність ефективних систем бронювання.

Тож предметною областю даного дослідження є розробка веб-сайту для бронювання робочих місць у коворкінгу з використанням рекомендаційної системи, який називається “CoworklySpace”.

Ось головні потреби для користувачів:

- коворкінг-простір надає клієнтам можливість орендувати робочі місця;
- надає переговорні кімнати та додаткові послуги на певний час;
- зручний спосіб бронювання та отримання їх рекомендацій щодо вибору місця та історії використання простору.

Системи рекомендацій є ключовими факторами підвищення зручності використання платформи, допомагаючи користувачам знаходити ідеальні місця для роботи на основі їхніх уподобань і бажань, а також історії бронювань і відгуків інших клієнтів. Така система сприяє оптимізації ресурсів, а також покращенню загального досвіду користувача.

Отже веб-сайт, який розробляється, повинен не тільки виглядати вимогам та сучасним стандартам створенню веб-сайтів, але ще й буде працювати на основі аналізу та вивчення поведінки користувачів, їхніх попередніх бронювань та оцінок.

1.2 Аналіз потенційних конкурентних переваг програмного продукту

На ринку просторів існують веб-сайти з коворкінгу, тому потрібно дізнатися та визначити плюси та мінуси існуючих аналогів.

1. OLX (рис. 1.1) - на платформі OLX доступні оголошення про оренду робочих місць у коворкінгу по всій Україні.

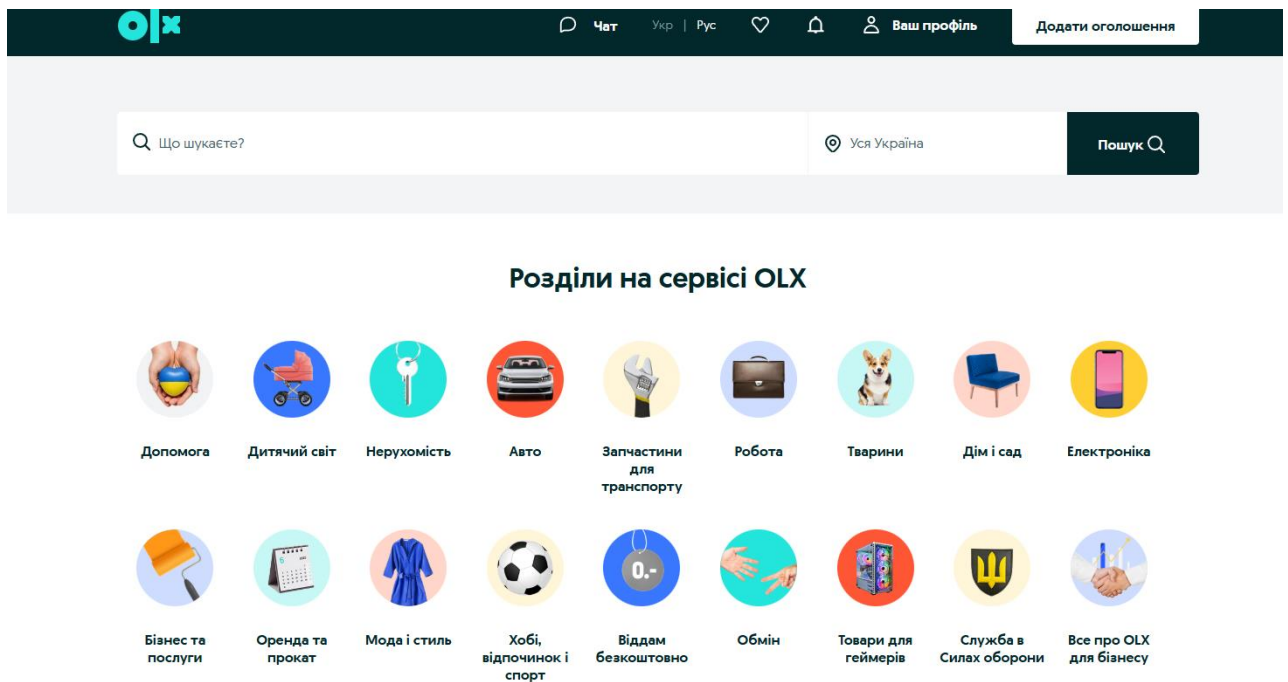


Рисунок 1.1 - Вебсайт "OLX"

Джерело – [2]

Переваги:

- великий вибір пропозицій – на платформі можна знайти багато варіантів коворкінгу по всій Україні;
- гнучкість у переговорах – користувачі можуть домовлятися про умови оренди безпосередньо з власниками;
- безкоштовне розміщення оголошень – для власників коворкінгів це зручний спосіб рекламувати свої послуги.

Недоліки:

- низька перевіреність оголошень – можливі шахрайські схеми або неактуальні пропозиції;
- відсутність бронювання онлайн – немає автоматизованого процесу оренди через платформу;
- неінтуїтивний пошук для коворкінгу – OLX не спеціалізується на бронюванні офісів, тому знайти відповідне робоче місце може бути складно;
- відсутність відгуків і рейтингової системи – користувачі не можуть залишати детальні оцінки сервісу.

2. Work-Talk (рис 1.2.) - об'єднує різноманітні коворкінги та робочі простори, дозволяючи користувачам бронювати робочі місця, переговорні кімнати чи конференц-зали через мобільний додаток.

Переваги:

- автоматизована система бронювання – можливість бронювати робочі місця та переговорні кімнати онлайн;
- корпоративний кабінет – зручне рішення для компаній, що дозволяє централізовано керувати бронюваннями;
- інтеграція з календарем – можливість синхронізації бронювань із Google Calendar.

Недоліки:

- менший вибір локацій – платформа працює з обмеженою кількістю коворкінгів;
- відсутність гнучких тарифів – більшість варіантів бронювання обмежується стандартними пакетами без погодинної оренди;
- мало користувачів на ринку України – сервіс ще не дуже популярний, що може впливати на вибір користувачів.

WorkIT (рис 1.3) - пропонує оренду офісів та робочих місць з сучасною інфраструктурою, включаючи автономне опалення та потужний генератор.

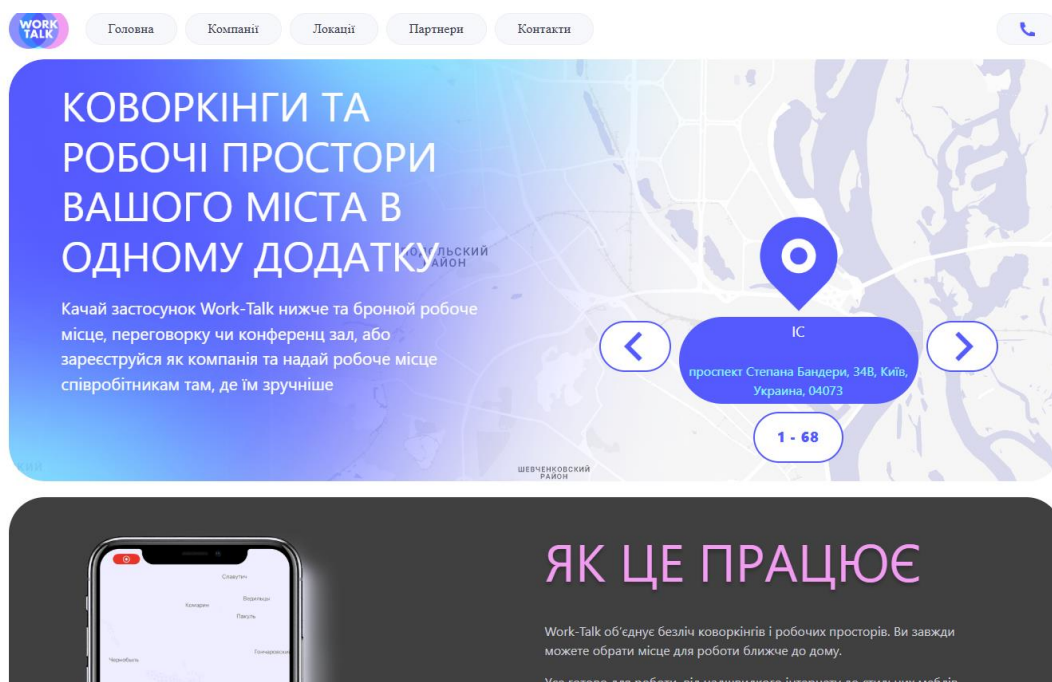


Рисунок 1.2 - Вебсайт "Work-Talk"

Джерело – [3]

Переваги:

- Зручне розташування – офіс розташований у Києві з гарною інфраструктурою.
- Повна автономність – є резервне живлення, генератори, що забезпечують безперебійну роботу.
- Гнучкі тарифи – можливість оренди погодинно, подово чи на тривалий термін.

Недоліки:

- Обмежене географічне охоплення – поки що доступний лише у Києві.
- Немає централізованої онлайн-платформи – користувачі змушені бронювати місця через адміністраторів або сторонні канали.
- Відсутність рекомендаційної системи – немає персоналізованого підбору робочого місця.

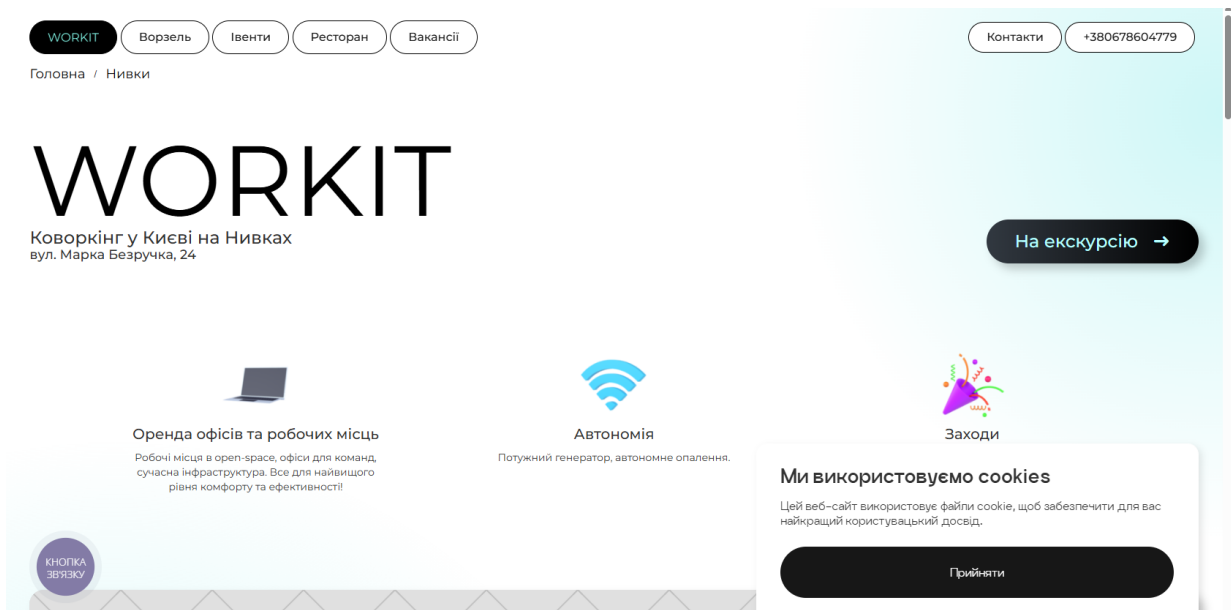


Рисунок 1.3 - Вебсайт “WorkIT”

Джерело – [4]

Для того щоб створений веб-сайт “CoworklySpace” мав конкурентні переваги на українському ринку, необхідно врахувати та додати особливості існуючих рішень і виділити ключові переваги та плюси.

- Індивідуальна рекомендаційна система – на відміну від багатьох існуючих сайтів, система буде використовувати алгоритми ШІ для персоналізованих пропозицій місць.
- Інтеграція з популярними сервісами – можливість синхронізації з Google, Telegram-ботами та мобільними додатками для зручного керування бронюваннями.
- Відгуки та рейтингова система – користувачі зможуть залишати коментарі та оцінки, що допоможе іншим клієнтам краще підбирати за своїми бажаннями.

Додання всіх цих функцій в веб-сайт забезпечить користувачам зручність в підбиранні робочих місць з коворкінгу та зробить це все більш комфортним.

1.3 Постановка завдання на кваліфікаційну роботу

Розробка веб-сайту з коворкінгу “CoworklySpace” веде за собою створення та реалізацію кращого вибору для користувачів робочих місць з коворкінгу та використання рекомендаційної системи. Створення приємного дизайну, зручного веб-додатку та ефективність його у використанні.

Ось завдання для виконання:

- аналіз вимог – дослідження ринку, виявлення потреб користувачів та визначення основних функцій системи;
- проєктування архітектури – створення структури бази даних, моделювання основних компонентів сайту та рекомендаційної системи для зручності використання;
- розробка функціоналу – створення веб-інтерфейсу, інтеграція системи бронювання, впровадження алгоритмів рекомендацій для покращення пошуку користувачам;
- тестування– перевірка роботи системи, виправлення помилок та покращення користувацького досвіду;
- розгортання та підтримка – запуск веб-сайту, забезпечення його стабільної роботою та оновлення функціоналу з часом.

Висновки до розділу 1

Тож за всіма критеріями було проведено аналіз та дослідження в якому було виявлено та сформовано ціль створення веб-сайту.

Опис предметної області дає чітку ціль та проблеми таких веб-застосунків, вказано їх основні проблеми та плюси. Після цього вже створено найкращі переваги веб-сайту “CoworklySpace”.

Крім того, сформульовано завдання для кваліфікаційної роботи, що включає в себе ключові етапи створення веб-сайту. Виконання цих етапів сприятиме розробці сучасного та функціонального продукту, який відповідатиме потребам цільової аудиторії.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ВЕБ-САЙТУ ДЛЯ БРОНЮВАННЯ РОБОЧИХ МІСЦЬ У КОВОРКІНГУ

2.1 Моделювання поведінки продукту

Перед безпосередньою розробкою веб-сайту для бронювання робочих місць у коворкінгу з використанням рекомендаційної системи було проведено попереднє моделювання поведінки майбутнього продукту. Це дозволить чітко уявити основні сценарії взаємодії користувачів із системою, а також визначити ключові компоненти, які повинні бути реалізовані на етапі проєктування та програмування. У даному розділі ми розглянемо побудову структури сайту «CoworklySpace».

У центрі уваги - користувач, який заходить на сайт із конкретною метою: знайти зручне робоче місце на певну дату та час, дізнатись про доступні зони, а також отримати рекомендації на основі своїх попередніх виборів чи вподобань. Сайт повинен бути інтуїтивно зрозумілим, працювати швидко та не викликати зайвих питань у процесі взаємодії. Це все допоможе не лише реалізувати зручний та функціональний сайт в межах кваліфікаційної роботи а й створити передумови для розширення функціоналу в майбутньому.

Функціональна структура сайту включає в себе:

- головна сторінка, яка демонструє роботу сайту, містить в собі майже весь функціонал, а саме: можливість обирати простори, рекомендації по місту та слайдер з відгуками користувачів, можливість авторизації та реєстрації користувача на сайті;
- модальні вікна кожного простору, ціни та зручності, бронювання та вибір дати та часу;
- сторінка з вибором просторів, сортування по місту та зручностям;
- головний кабінет користувача, можливість зміни пароля та пошти, заброньовані простори та можливість їх видалення;

- сторінка відгуків, можливість написання відгуку та виставлення оцінки;
- сторінка опису веб-сайту та компанії.

В основі побудови структури сайту – принцип SPA (Single Page Application) [5], що забезпечується використанням JavaScript. Це означає, що перехід між розділами відбувається без повного перезавантаження сторінки, що забезпечує більшу швидкість та зручність користування.

Схематична структура компонентів вебсайту:

- Header (логотип, меню, навігація, авторизація);
- Main (головна, бронювання, профіль);
- Body (тіло веб-сторінки);
- Section (список доступних місць, рекомендації, інформація);
- Div (організація вмісту);
- Footer (копірайт);
- Script(перевірка форм, динамічне оновлення контенту).

Кожен з компонентів реалізований у вигляді HTML-блоку з відповідним CSS-оформленням та динамічною логікою на JavaScript.

Проектування навігації:

- меню навігації дозволяє переміщатись між головною сторінкою, головним кабінетом користувача, вибором простору, а також сторінкою опису веб-сайту. Меню є адаптивним – при зміні розміру екрана воно трансформується у випадаючий список (бургер-меню).

Формування рекомендаційної системи:

- блок рекомендацій просторів формується за вибором міста, тобто користувач натискає на модальне вікно і система запам'ятовує місто користувача

Це реалізується через локальне збереження переглядів у LocalStorage.

Проектування авторизації.

- дозволяє користувачу залишатися у своєму кабінеті та користуватися бронюванням просторів. Реалізація авторизування відбувається через систему Firebase, збереження користувача у системі.

Діаграма прецедентів [6] (рис. 2.1) є важливим інструментом моделювання функціональності веб-сайту з бронювання коворкінгу CoworklySpace, а також взаємодії користувачів із його основними модулями. Такий тип діаграм використовується для візуалізації ролей користувачів та їхніх можливих дій у межах системи.

На діаграмі зображено актора (модель) – користувача, який взаємодіє з системою через головну сторінку сайту. Головна сторінка забезпечує доступ до таких функцій як:

- реєстрація або авторизація облікового запису;
- перехід у кабінет користувача;
- перехід у вибір та бронювання просторів.

У свою чергу, вибір просторів передбачає два сценарії:

- вибір простору через сторінку просторів, яка відкриває модальні вікна з ціною, описом та зручностями, вибір за містом;
- вибір просторів за допомогою рекомендаційної системи на головній сторінці, яка включає в себе вибір за містом.

Кожен прецедент («реєстрація», «авторизація», «вибір простору», «написання відгуку», «бронювання простору») є моделлю поведінки користувача, що описує конкретний функціональний випадок використання системи.

Актором у діаграмі виступає користувач, який ініціює взаємодію із сайтом. Зв'язки між актором та прецедентами показують, які дії є доступними для нього на різних етапах використання ресурсу.

Таким чином, діаграма прецедентів дозволяє узагальнено представити логіку роботи веб-сайту та служить основою для формалізації вимог до його функціональних компонентів.



Рисунок 2.1 - Діаграма прецедентів

Джерело: розроблено автором

Діаграма діяльності [7] (рис. 2.2) є одним із основних інструментів візуалізації логіки виконання процесів у системі продукту. У рамках мого проєкту – веб-сайту для бронювання місць у коворкінгу з використанням рекомендаційної системи – вона дозволяє чітко показати послідовність дій користувача та системи, починаючи від авторизації до завершення бронювання простору.

Тобто умовно весь процес взаємодії починається з взаємодії користувачем головної сторінки сайту. Далі є дві можливі гілки шляху:

Перехід на сторінку вибору простору з використанням фільтру міста та зручностей;

Перегляд рекомендованих просторів на основі історії перегляду за містом.

Під час вибору простору можливий огляд його інформації, ціни, зручностей та міста в якому він знаходиться. Далі можливість бронювання на окремий час та дату і додавання його у кабінет користувача. У разі не авторизованого облікового запису користувач виконує реєстрацію на сайті для збереження рекомендацій та його кабінету.

Тобто діаграма діяльності відображає основні етапи взаємодії користувача з платформою, дає взаємозв'язки з різними сценаріями та

забезпечує гнучкість, логічність, зручність користувацького досвіду на веб-сайті.

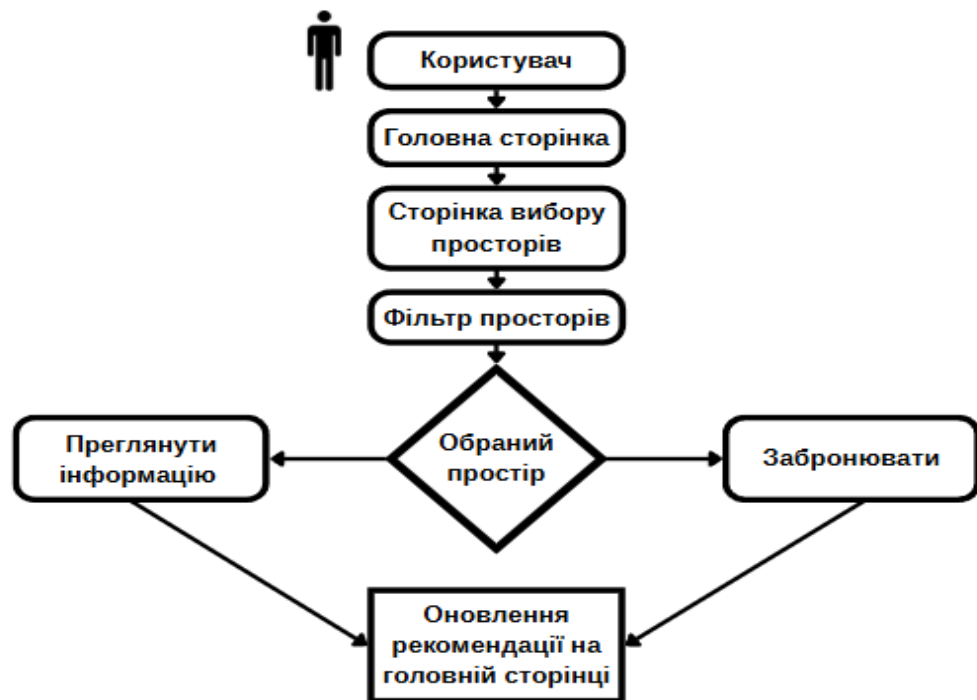


Рисунок 2.2 - Діаграма діяльності

Джерело: розроблено автором

Діаграма послідовності[8] (рис. 2.3) ілюструє, як об'єкти у системі взаємодіють між собою в певному порядку часу.

Сценарій на веб-сайт CoworklySpace виглядає наступним чином:

1. Користувач відкриває сторінку вибору простору.
2. Інтерфес надсилає запит на завантаженні повного каталогу прострів
3. Користува обирає параметри фільтрів (зручності, місто).
4. Веб-сайт зберігає ці дані та виконує пошук просторів за фільтром.
5. Користувач натискає на обраний ним простір та відкриває модальне вікно з детальним описом.
6. Дані з вибору міста зберігаються у локальне сховище (Local Storage).

7. На головній сторінці веб-сайту тепер у вікні відображається рекомендаційна система з просторами того міста, на яке натиснув користувач.

8. Далі користувач додає обраний простір з вибором дати та часу у свій особистий кабінет, де і зберігаються бронювання.

Діаграма послідовності дозволяє відстежити не лише порядок дій користувача, а й логіку обміну даними між модулями самого веб-сайту.

Це важливо для реалізації клієнтської логіки та подальшого розширення функціоналу веб сайту у майбутньому.

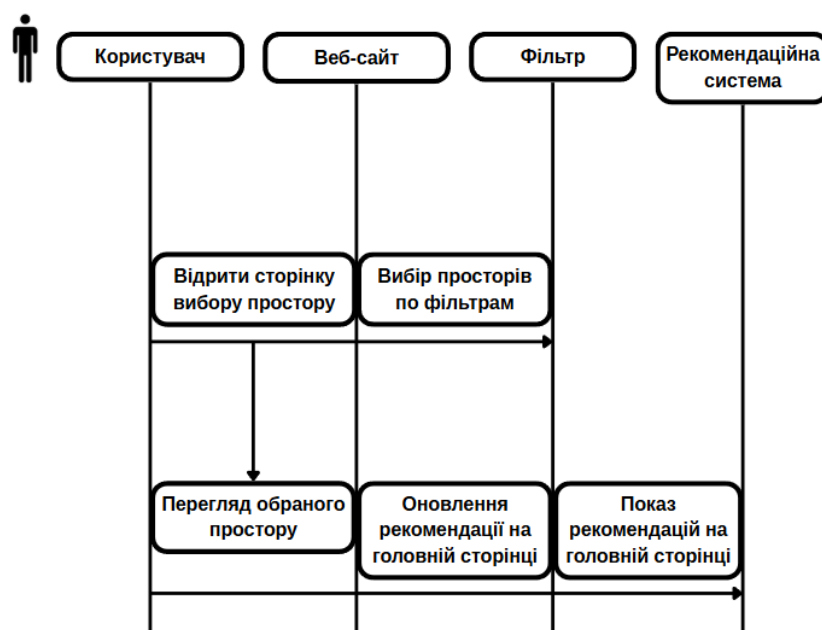


Рис. 2.3 - Діаграма послідовності

Джерело: розроблено автором

Діаграма комунікації[9] (рис. 2.4) показує, як об'єкти взаємодіють між собою в рамках конкретного сценарію, зосереджуючись на структурі зв'язків та порядку повідомлень веб-сайту CoworklySpace.

Тобто користувач ініціює дії, які обробляються різними функціями та компонентами системи.

Основні компоненти такі:

Авторизація FireBase – дозволяє швидко та безпечно реалізувати вхід у систему за допомогою електронної пошти або Google-акаунта. Це значно

спрощує процес входу для користувача і забезпечує захист персональних даних.

Головний кабінет користувача – забезпечує персональний простір в якому зібрана інформація про користувача, його попередні та активні бронювання, відгуки та налаштування профілю.

Фільтри вибору просторів – допомагає швидко знайти відповідне робоче місце, виходячи з параметрів: розташування та зручності.

Рекомендаційна система – надає на основі попередніх бронювань користувача персоналізовані рекомендації та використовує збережену історію переглядів. Отримує дані локального сховища (Local Storage).

LocalStorage – використовується для зберігання інформації про переглянуті простори. Кожен раз, коли користувач переглядає простір та додає його у бронювання, ця інформація записується у браузер користувача для подальшого використання

Система відгуків – дає змогу користувачам залишити відгук після використання простору. Інші користувачі можуть переглядати ці відгуки, що створює довіру та сприяє вибору якісного місця.

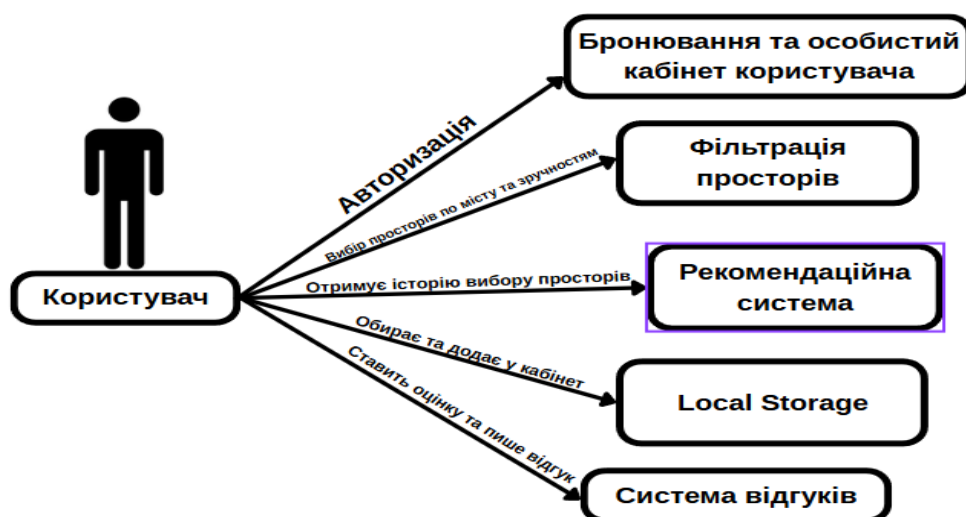


Рисунок 2.4 – Діаграма комунікації

Джерело: розроблено автором

2.2 Моделювання структури продукту

Моделювання структури програмного продукту дозволяє візуалізувати, як саме побудована система зсередини – з яких об'єктів вона складається, які між ними зв'язки, і як це все організовано з точки зору архітектури. У цьому підрозділі розглядаються ключові структурні діаграми сайту CoworklySpace згідно з мовою моделювання UML: діаграма класів, діаграма об'єктів та діаграма пакетів.

До основних діаграм, що дозволяють описати структуру програмного продукту, належать:

- діаграма класів;
- діаграма об'єктів;
- діаграма пакетів.

Кожна з них слугує інструментом для моделювання різних аспектів системи: класів та їхніх зв'язків, конкретних об'єктів у момент часу, структурних залежностей між модулями.

Діаграма класів (Class Diagram) [10] (рис. 2.5)

Діаграма класів відображає основні логічні одиниці системи – класи, їхні атрибути, методи та взаємозв'язки. У проєкті веб-сайту CoworklySpace було виділено кілька ключових класів:

- User – містить інформацію про користувача, включаючи ID, ім'я, email, роль (адмін/звичайний);
- CoworkingSpace – описує простір: назва, розташування, рейтинг, список зручностей;
- Booking – зберігає дані про конкретне бронювання: дата, час, ID користувача, ID простору;
- Review – модель відгуку з текстом, рейтингом та автором;
- RecommendationEngine – окремий клас, який обробляє логіку рекомендацій на основі історії користувача.

Між класами встановлено відносини:

- асоціація між User і Booking (один користувач - багато бронювань);
- агрегація між CoworkingSpace і Review (один простір - багато відгуків);
- використання RecommendationEngine з боку User.

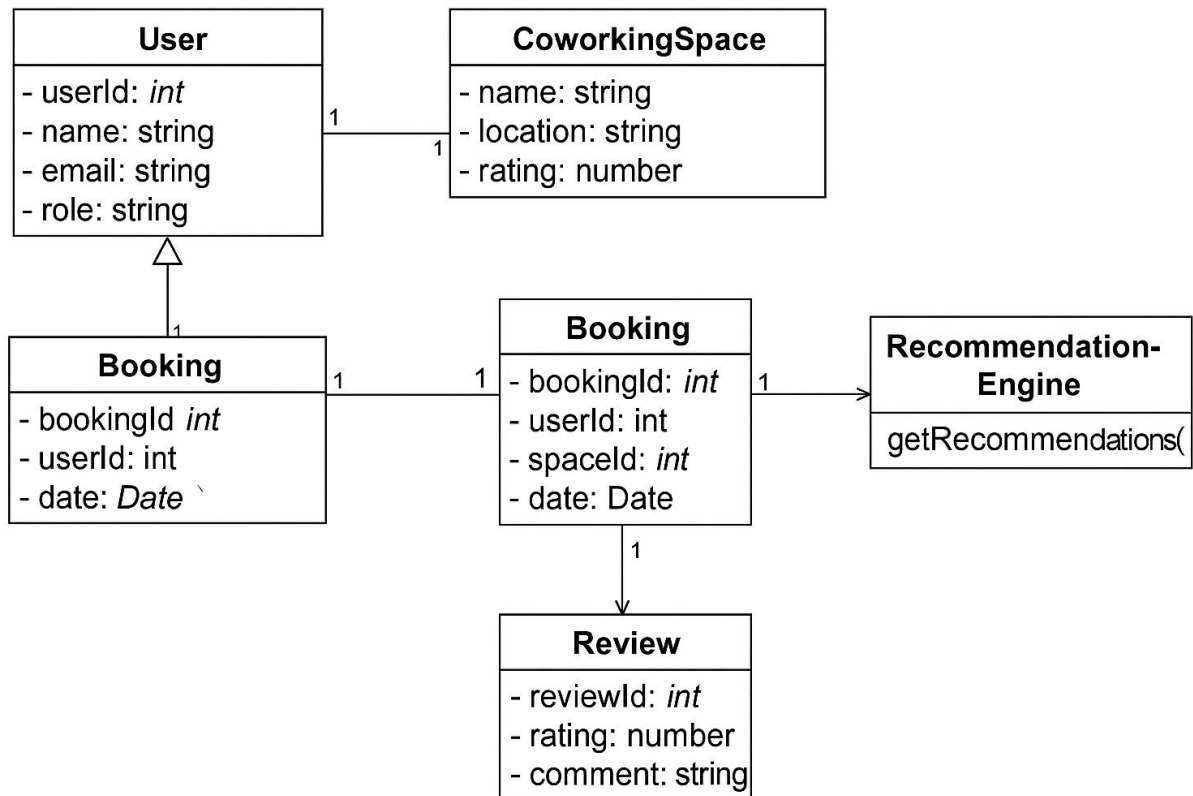


Рисунок 2.5 – Діаграма класів

Джерело: розроблено автором

Діаграма об'єктів (Object Diagram) [11] (рис. 2.6)

На основі діаграми класів створюється діаграма об'єктів, яка демонструє конкретні екземпляри класів у певний момент часу. Для сайту CoworklySpace така діаграма може демонструвати, наприклад, те що:

На сторінці просторів є об'єкти `coworkings`

Користувач відкриває модальне вікно `modal` з детальною інформацією про простір

У локальне сховище зберігається місто в якому знаходиться цей простір

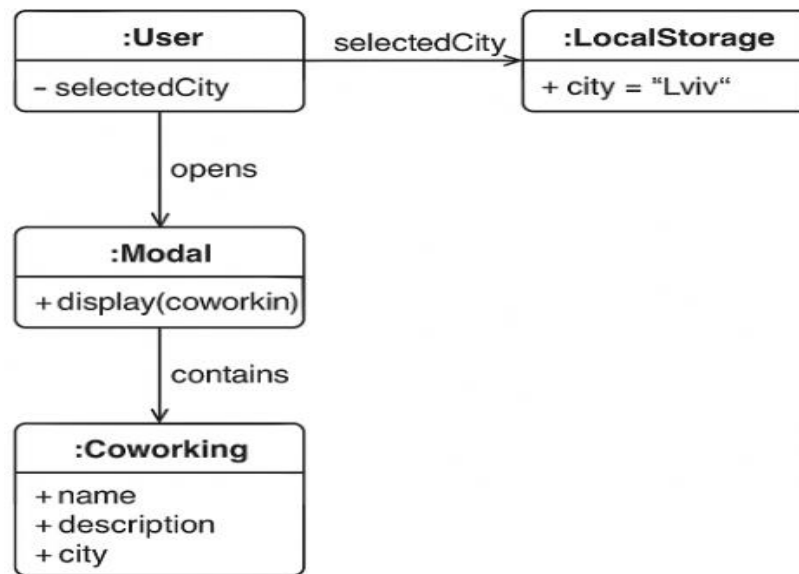


Рисунок 2.6 – Діаграма об'єктів

Джерело: розроблено автором

Діаграма пакетів (Package Diagram) [12] (рис. 2.7)

Для впорядкування структури проєкту на логічні частини використовується діаграма пакетів. Вона демонструє, як компоненти згруповані у модулі, і які між ними залежності. Основні пакети в застосунку веб-сайту CoworklySpace такі:

- `auth` – модулі для реєстрації, входу та перевірки сесій (інтеграція з Firebase).
- `booking` – логіка бронювання, перевірка доступності, створення записів.
- `spaces` – компоненти для перегляду, сортування та фільтрації просторів.
- `reviews` – модулі відгуків, рейтингів та їх виводу.
- `recommendation` – логіка персоналізованих рекомендацій.
- `context` – глобальний стан користувача та взаємодія з LocalStorage.

Діаграма дозволяє візуалізувати, які модулі залежать один від одного, і сприяє майбутньому масштабуванню системи.

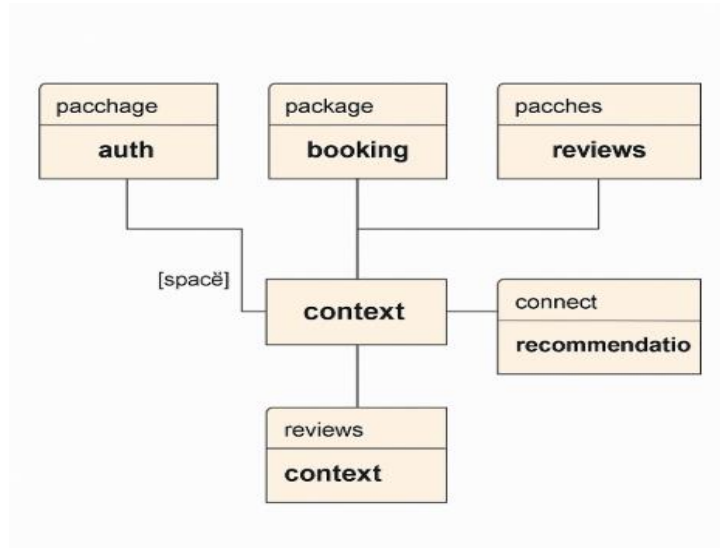


Рисунок 2.7 – Діаграма пакетів

Джерело: розроблено автором

Таким чином, структурне моделювання дозволяє формалізувати логіку побудови веб-сайту, описати його архітектурні блоки, залежності між компонентами та типи даних, що використовуються. Це є основою для подальшого обслуговування, оптимізації та масштабування програмного продукту.

2.3 Опис архітектури продукту

Архітектура веб-застосунку CoworklySpace побудована на основі сучасних принципів розробки веб-додатків, забезпечуючи модульність, масштабованість та зручність у підтримці. Основна мета архітектури – створення інтуїтивно зрозумілого інтерфейсу для користувачів, що дозволяє ефективно взаємодіяти з просторами коворкінгів.

Тип архітектури: Клієнт-серверна модель, тобто лієнт (браузер користувача): відповідає за інтерфейс, взаємодію з HTML-сторінками, відображення контенту та збереження стану у localStorage.

Архітектурна модель

1. Рівень представлення (Presentation Layer)

Цей рівень відповідає за взаємодію з кінцевим користувачем. Основним його завданням є відображення даних, оформлення інтерфейсу, а також передача подій до рівня логіки. Інтерфейс реалізовано за допомогою HTML-розмітки та CSS-стилів. Додатково використовується JavaScript для реалізації динамічної поведінки інтерфейсу.

Користувач взаємодіє з елементами сторінок, зокрема:

- натискає на картки коворкінгів;
- використовує модальне вікно для перегляду інформації;
- вводить дані у поле пошуку;
- обирає місто для фільтрації результатів.

Ці дії ініціюють виклики відповідних обробників подій у JavaScript, що забезпечує зв'язок з рівнем логіки.

Взаємодія відбувається коли користувач натискає кнопку, далі інтерфейс реагує через JavaScript та викликається функція з рівня логіки.

2. Рівень логіки (Business Logic Layer)

Рівень логіки визначає поведінку застосунку, обробляє події, керує взаємодією між користувачем і системою. Основні функції реалізовано в окремих JavaScript-файлах, що забезпечує модульність та зручність супроводу.

До основних функцій рівня логіки належать:

- завантаження та відображення списку коворкінгів;
- обробка кліку на елементи та відкриття модального вікна;
- пошук простору за ключовими словами;
- збереження вибраного міста у локальному сховищі;
- динамічне оновлення відображеного списку залежно від параметрів фільтрації.

Логіка побудована таким чином, що дані не завантажуються ззовні, а працюють із внутрішніми масивами або JSON-структурами. У разі розширення проєкту можлива інтеграція з API для роботи з базою даних.

3. Рівень даних (Data Layer)

Цей рівень відповідає за зберігання, отримання та оновлення даних. У поточній версії застосунку він реалізований локально, але структура передбачає можливість підключення до бази даних у майбутньому.

Типи даних:

- статичні JSON-дані або масиви з інформацією про коворкінги;
- localStorage – зберігання обраного міста, даних авторизації (у простому вигляді).

Такий підхід дозволяє уникнути створення окремого сервера та зосередитись на розробці логіки застосунку на стороні клієнта.

У структурі продукту виділено такі основні частини:

- index.html – основний файл, з якого починається завантаження інтерфейсу. Містить елементи інтерфейсу: заголовки, контейнер для рекомендацій, тощо;
- style.css – таблиця стилів, відповідальна за зовнішній вигляд сторінок: сітку, кольори, шрифти, розміщення елементів;
- search.html – файл, що містить функції для динамічного формування вмісту вибору просторів на сторінці та реалізації фільтрів для детального вибору простору для покращення пошуку;
- js/index-auth.js – містить конфігурацію та ініціалізацію Firebase-з'єднання з базою даних;
- dashboard.html – файл, що містить функції для особистого кабінету користувача, налаштування його даних та місця для бронювання просторів;
- js/booking/js – містить конфігурацію для зберігання просторів та бронювання їх у кабінет користувача.

Модель Крухтена 4+1.

Для системного проектування веб-застосунку CoworklySpace було використано модель архітектурного представлення «4+1» за Крухтеном. Цей підхід передбачає опис архітектури програмного продукту з п'яти різних точок зору: логічної, процесної, фізичної, розробницької та сценарної. Така модель

дозволяє отримати комплексне уявлення про структуру та функціонування системи.

1. Логічне представлення (Logical View)

Цей рівень описує основні функціональні можливості системи з точки зору кінцевого користувача. У веб-застосунку CoworklySpace логічна структура побудована на основі взаємодії між модулями, які відповідають за відображення, фільтрацію, вибір та перегляд коворкінг-просторів.

2. Процесне представлення (Process View)

Цей рівень описує динаміку виконання системи та взаємодію її компонентів у процесі роботи.

Система працює асинхронно, що дозволяє уникнути перезавантаження сторінок і забезпечити кращу продуктивність.

3. Розгортання (Deployment View)

Рівень розгортання описує фізичне розміщення компонентів системи. CoworklySpace є клієнтським веб-застосунком, що розгортається як статичний сайт на хостингу (наприклад, GitHub Pages). Усі файли HTML, CSS, JavaScript і зображення зберігаються у публічному доступі і завантажуються у браузері користувача.

4. Представлення розробника (Development View)

Цей рівень демонструє, як система організована з точки зору програміста. Код застосунку поділений на окремі модулі за функціональним призначенням. Структура проєкту є логічно впорядкованою:

5. Сценарії використання (Use Case View)

Цей рівень описує ключові сценарії, у межах яких користувач взаємодіє із системою. До основних сценаріїв входять:

Ця модель забезпечує повне уявлення про архітектурну будову та динаміку функціонування веб-застосунку CoworklySpace з точки зору розробки, користування та розгортання.

Висновки до розділу 2

У другому розділі було змодельовано поведінку вебсайту для бронювання робочих місць у коворкінгу з використанням рекомендаційної системи

Було визначено головні функціональні складові проєкту, зосереджено увагу на зручності користувача та оптимальній структурі сторінок.

Реалізовано підхід SPA, який забезпечує швидкість та комфорт у використанні сайту

Розроблено навігацію адаптовану під різні пристрої продумано структуру основних сторінок таких як головна сторінка, кабінет користувача сторінка, вибору простору та відгуків

Створено блок рекомендацій на основі LocalStorage що дозволяє персоналізувати досвід користувача

Впроваджено авторизацію через FireBase для збереження облікових даних та бронювань

Побудовано діаграми прецедентів діяльності послідовності та комунікації які відображають логіку функціонування сайту на різних етапах взаємодії з користувачем.

Завдяки цьому було сформовано цілісне бачення вебсайту CoworklySpace як функціонального та зручного сервісу для бронювання місць у коворкінгу з інтелектуальними підказками та гнучкою системою роботи.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ДЛЯ БРОНЮВАННЯ РОБОЧИХ МІСЦЬ У КОВОРКІНГУ

3.1 Реалізація та конструювання програмного продукту

Весь дизайн створювався безпосередньо в кодї, з використанням HTML, CSS та JavaScript. Цей підхід дозволив одночасно розробляти як візуальну частину, так і функціональність сайту рис(3.1)

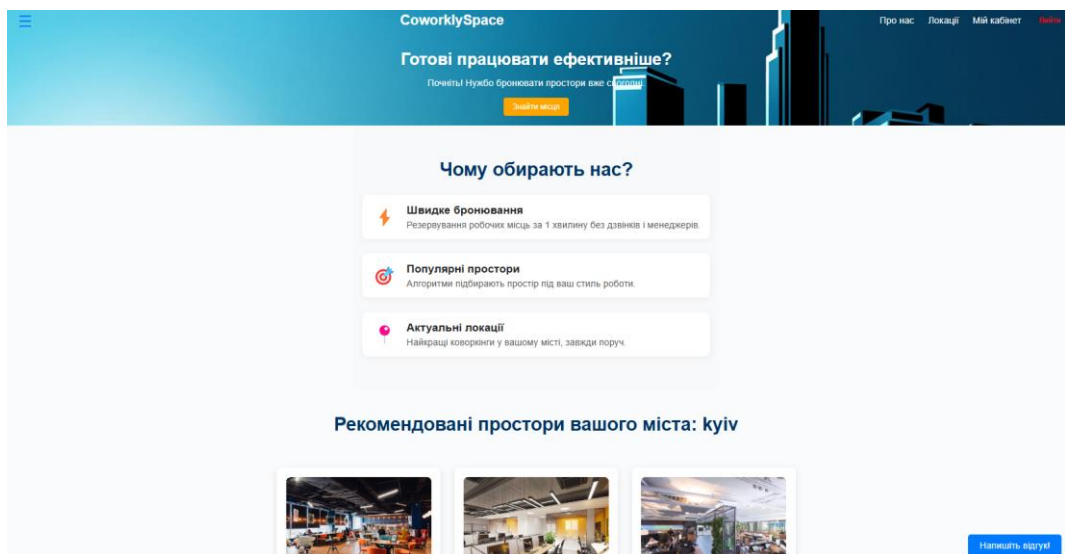


Рисунок 3.1 - Головна сторінка CoworklySpace

Джерело: розроблено автором

Створення високоякісного прототипу - високоякісний прототип реалізовано безпосередньо в кодї. Основні сторінки, такі як index.html, dashboard.html, search.html, auth.html та reviews.html, були створені з урахуванням логіки користувацьких сценаріїв. Це дозволило швидко тестувати та вдосконалювати інтерфейс без додаткових інструментів.

Обґрунтування колористики - у проєкті використано стриману та сучасну кольорову палітру. Основні кольори: білий для фону, темно-сірий для

тексту та акцентний синій для інтерактивних елементів. Це забезпечує високий контраст та зручність сприйняття інформації.

Опис типографіки - для текстових елементів використано шрифт Inter, який забезпечує хорошу читабельність на різних пристроях. Розміри шрифтів адаптовані для різних типів контенту: заголовки – 24–32 pt, основний текст – 16 pt, підписи – 12 pt.

Інтерактивний прототип - інтерактивність реалізовано за допомогою JavaScript. Функціональні елементи, такі як форма авторизації (auth.html), пошуковий фільтр (search.html) та система рекомендацій, забезпечують динамічну взаємодію з користувачем.

Для покращення видимості сайту в пошукових системах впроваджено базові принципи SEO:

- Семантична розмітка HTML5
- Оптимізовані мета-теги (title, description)
- Використання alt-тегів для зображень
- ЧПУ (людинозрозумілі URL)
- Адаптивний дизайн для мобільних пристроїв

```
document.getElementById('searchInput').addEventListener('input',
function() {
    const query = this.value.toLowerCase();
    const results = coworkingSpaces.filter(space =>
space.name.toLowerCase().includes(query));
    displayResults(results);
});
```

Адаптивний прототип (desktop, mobile)

Дизайн адаптовано для різних розмірів екранів за допомогою медіа-запитів у CSS. Це забезпечує зручність користування сайтом на будь-якому пристрої.

```
@media (max-width: 768px) {  
  .nav-menu {  
    flex-direction: column;  
  }  
  .search-bar {  
    width: 100%;  
  }  
}
```

Це дозволяє користувачам зручно користуватися платформою як на десктопах, так і на мобільних пристроях.

3.2 Тестування програмного продукту

Тестування інтерфейсу на відповідність принципам доступності (WCAG 2.1)

На даному етапі була проведена перевірка відповідності інтерфейсу веб-додатку стандартам доступності згідно з рекомендаціями WCAG 2.1 (Web Content Accessibility Guidelines). Метою було забезпечити зручність використання платформи для якомога ширшого кола користувачів, включаючи людей із порушеннями зору чи моторики.

Було виконано наступне:

- **Контрастність тексту.** Весь текст на платформі має контраст щонайменше 4.5:1 (для основного тексту) і 3:1 (для великих заголовків), що відповідає критерію рівня AA. Для перевірки використовувався онлайн-інструмент WebAIM Contrast Checker.

- **Альтернативні описи до зображень.** Усі значущі графічні елементи (іконки, фото коворкінгів) мають атрибут alt, наприклад:

```

```

- Фокус і клавіатурна навігація. Усі основні елементи (меню, форми, кнопки) доступні з клавіатури за допомогою клавіші Tab. Стили `:focus-visible` були додані вручну:

```
button:focus-visible,
input:focus-visible {
  outline: 2px solid #18BC9C;
  outline-offset: 2px;
}
```

- Озвучування елементів (екранні рідери). Інтерфейс проходив тестування за допомогою NVDA (екранний рідер для Windows). Усі кнопки мали зрозумілі `aria-label`:

```
<button aria-label="Додати відгук">+</button>
```

- Мобільна доступність. Перевірено, що масштабування працює на рівні 200% без втрати функціональності чи зручності.

Також було протестовано сайт за допомогою розширення **axe DevTools**, що виявило лише незначні помилки (наприклад, дублікати `id`, які були усунені під час фінального редизайну).

3.3 Використання програмного продукту

Після завершення етапу проєктування інтерфейсу та розробки дизайну було реалізовано повноцінну функціональність веб-сайту CoworklySpace. Основною метою створення даного ресурсу було забезпечення зручної та інтуїтивної платформи для бронювання робочих місць у коворкінгах чотирьох українських міст: Києва, Харкова, Львова та Одеси.

Використання сайту починається з головної сторінки (`index.html`), де користувач знайомиться з основною інформацією про проєкт та може перейти до реєстрації чи входу через кнопку "Увійти / Реєстрація". Аутентифікація

реалізована за допомогою Firebase Authentication, що забезпечує безпечний і швидкий вхід у систему за допомогою email і пароля. У разі успішного входу інтерфейс динамічно змінюється: кнопка входу зникає, а на її місці відображається електронна пошта користувача. Також з'являється особистий кабінет користувача (рис. 3.2)

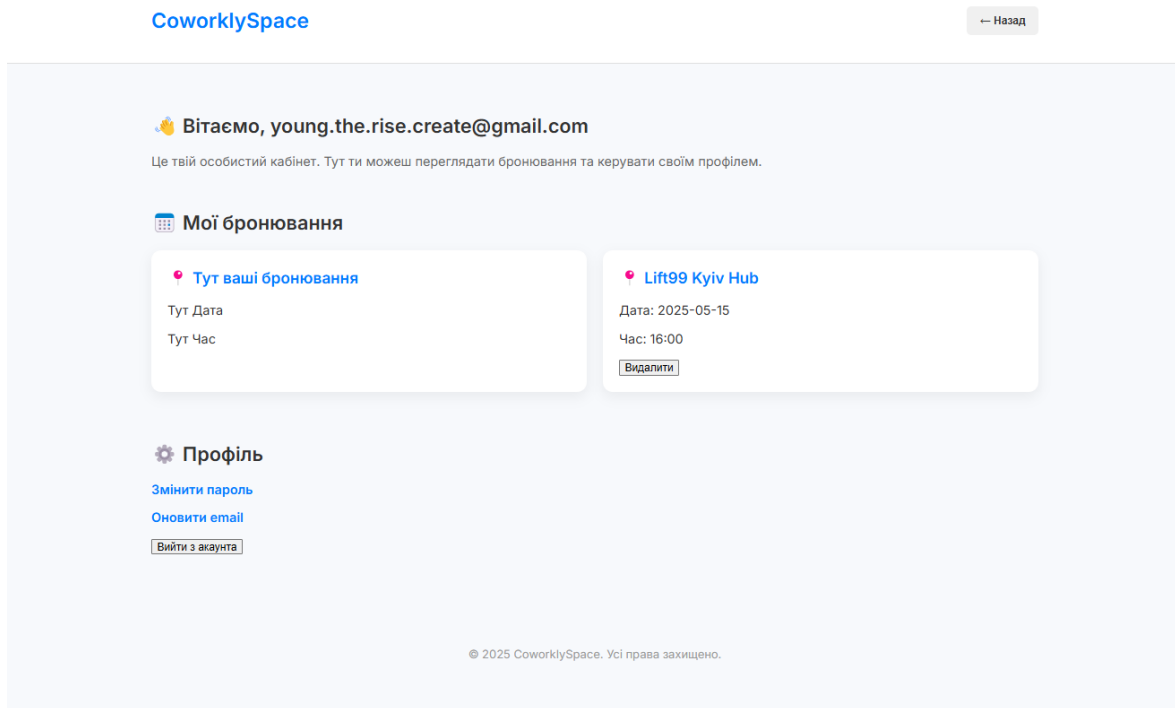


Рисунок 3.2 - Кабінет користувача CoworklySpace

Джерело: розроблено автором

Після входу користувач може переглядати доступні локації для бронювання в одному з чотирьох міст. Для цього реалізована навігація по сторінках із містами, де кожне місто має по три унікальні коворкінг-локації. Кожна локація має:

- короткий опис,
- фото,
- рейтинг (відгуки),
- кнопку для бронювання.

Бронювання реалізується як логічна дія (для навчального проєкту – без збереження у базі даних, але з імітацією переходу до наступного етапу). У повній версії проєкту це можна розширити зберіганням даних у Firebase Realtime Database або Firestore.

Користувач також має можливість залишити відгук. Відгуки надсилаються через форму, що зберігає дані в Firebase, а потім динамічно відображає їх на сторінці. Відгуки пов'язані з обліковим записом, що дає змогу ідентифікувати автора.

Окремо варто відзначити адаптивність сайту. Завдяки використанню медіа-запитів (@media) у CSS та гнучких сіток (flex, grid), інтерфейс коректно відображається на мобільних пристроях, планшетах та десктопах. Це особливо важливо, оскільки більшість користувачів здійснюють пошук і бронювання саме з мобільних телефонів.

Приклад адаптивного фрагмента CSS:

```
@media (max-width: 768px) {  
  .coworking-card {  
    flex-direction: column;  
    align-items: center;  
  }  
}
```

Загалом, використання веб-сайту є зручним, логічно організованим і не потребує додаткового навчання для користувача. Простота дизайну, швидкодія і доступність функцій роблять CoworklySpace зручним інструментом для бронювання коворкінгів навіть для нових відвідувачів.

На рис 3.3 зображено функцію фільтрації на сторінці вибору просторів. Також кожен простір має модальне вікно та кнопку бронювання рис.3.4.

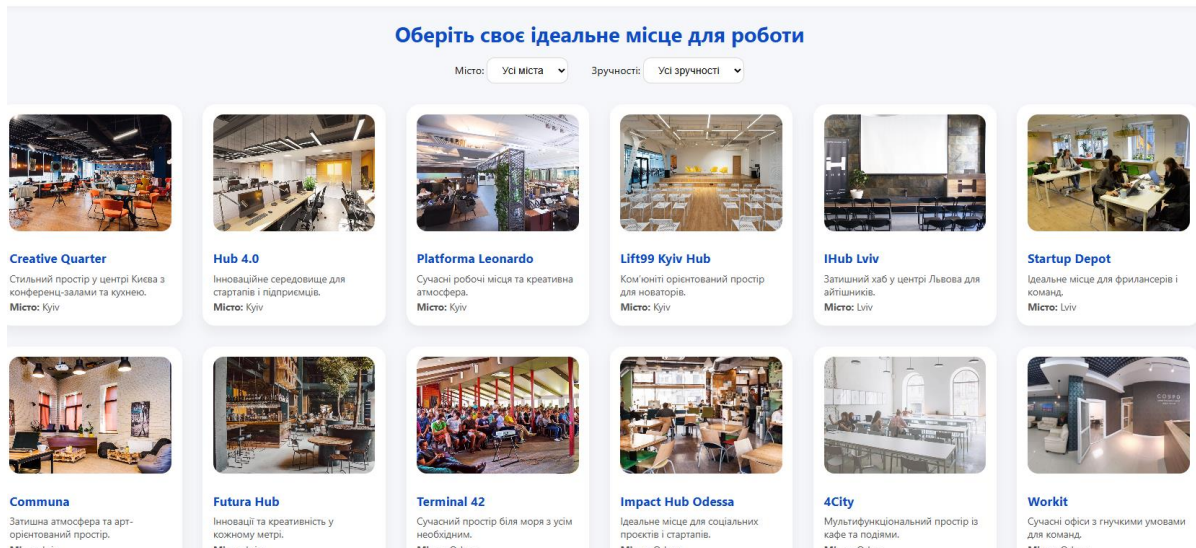


Рисунок 3.3 - Сторінка CoworklySpace з фільтрацією

Джерело: розроблено автором

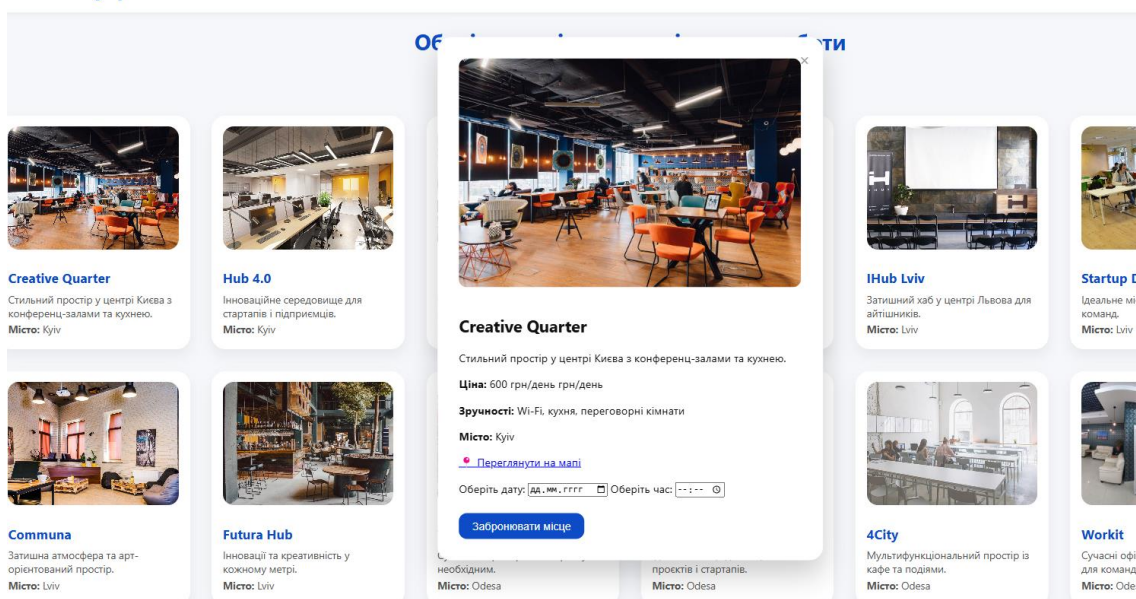


Рисунок 3.4 - Модальне вікно з вибором простору CoworklySpace

Джерело: розроблено автором

Висновки до розділу 3

У цьому розділі було детально розглянуто процес реалізації UI/UX дизайну платформи CoworklySpace – веб-додатку для пошуку та бронювання робочих місць у коворкінгах. Цей етап відіграє ключову роль у створенні

продукту, який не лише функціональний, а й зручний та привабливий для кінцевого користувача. Правильно спроектований інтерфейс сприяє інтуїтивному взаємодії з платформою, знижує поріг входу для нових користувачів та формує позитивне враження від користування сервісом.

У підрозділі 3.1 було здійснено повний цикл прототипування – від вибору кольорової гами та типографіки до створення інтерактивного прототипу з урахуванням адаптивного дизайну. Основний акцент було зроблено на простоті, контрастності та сучасному вигляді інтерфейсу. Колористика вибрана таким чином, щоб викликати довіру (темно-синій фон) і водночас мотивувати до дії (бірюзові акценти). Шрифт *Roboto* забезпечує високу читабельність на різних пристроях, що також відповідає вимогам адаптивності. Особливу увагу приділено SEO-оптимізації, що дозволить сайту бути краще індексованим у пошукових системах.

У підрозділі 3.2 проведено тестування інтерфейсу відповідно до міжнародних стандартів доступності WCAG 2.1. Були перевірені всі критичні аспекти: контрастність кольорів, наявність альтернативних текстів для зображень, коректна навігація з клавіатури, а також доступність для користувачів екранних рідерів. Завдяки застосуванню відповідних CSS-стилів і атрибутів ARIA, вдалося досягти високого рівня інклюзивності веб-інтерфейсу.

У підрозділі 3.3 була здійснена оцінка технічної та часової доцільності реалізації проєкту, а також підготовка всіх необхідних матеріалів для передачі: структурована документація, інтерактивні прототипи, файли стилів, гайдлайни та інструкції для розгортання. Це свідчить про завершеність і готовність дизайну до подальшої інтеграції та масштабування.

Загалом, розроблений інтерфейс відповідає сучасним вимогам до зручності, доступності та візуальної привабливості. Реалізація UI/UX дизайну стала фундаментом для успішної розробки всього проєкту. Досягнуто основну мету – створити інтерфейс, який гармонійно поєднує технічну простоту з користувацьким комфортом.

ВИСНОВКИ

У межах кваліфікаційної роботи було успішно реалізовано повний цикл створення веб-застосунку CoworklySpace, призначеного для бронювання робочих місць у коворкінгах із використанням персоналізованої рекомендаційної системи. Під час дослідження проведено глибокий аналіз предметної області, виявлено недоліки існуючих рішень та сформульовано вимоги до програмного продукту з урахуванням потреб цільової аудиторії.

У процесі проектування було змодельовано структуру системи, побудовано діаграми поведінки, структури та взаємодії користувачів із застосунком. Застосовано сучасні підходи до розробки SPA (Single Page Application), що забезпечило швидкодію та зручність навігації. Упроваджено функціональну рекомендаційну систему на основі локального збереження даних, яка надає користувачу індивідуалізовані пропозиції коворкінг-просторів.

У рамках реалізації веб-сайту створено адаптивний інтерфейс із дотриманням принципів доступності (WCAG 2.1), що гарантує комфортне використання як на десктопних, так і на мобільних пристроях. Проведено тестування функціональності та інтерфейсу, що підтвердило надійність і зручність платформи.

Таким чином, поставлену мету – створення сучасного вебсайту для бронювання робочих місць у коворкінгу з рекомендаційною системою – досягнуто. Розроблений продукт відповідає актуальним вимогам до веб-застосунків, вирізняється інтуїтивним інтерфейсом, адаптивністю, доступністю та високою практичною цінністю.

Отримані результати можуть бути використані для подальшого вдосконалення платформи: інтеграції з реальними базами даних, розширення географії доступних локацій, розробки мобільної версії та впровадження більш складних алгоритмів рекомендацій на основі штучного інтелекту.

ДОДАТОК А

ФРАГМЕНТИ ЛІСТИНГУ

```

const recommendedCity = localStorage.getItem('recommendedCity');
const spaceCardsContainer = document.getElementById('space-cards');

const spaces = [
  { name: 'Creative Quarter', city: 'Kyiv', location: '', rating: '★★★★☆', image: 'https://creativequarter.net/wp-content/uploads/2019/08/location-gulliver_new.jpg'},
  { name: 'Hub 4.0', city: 'Kyiv', location: '', rating: '★★★★★', image: 'https://4hub.com.ua/wp-content/uploads/2019/10/smart-office-14.jpg'},
  { name: 'Platforma Leonardo', city: 'Kyiv', location: '', rating: '★★★★☆', image: 'https://static.wixstatic.com/media/09d8dc_3594a39862264187bc33050d4b89a20d-mv2.jpg/v1/f'},
  { name: 'IHub Lviv', city: 'Lviv', location: '', rating: '★★★★★', image: 'https://ihub.ua/wp-content/uploads/2022/09/lv_ev_10-min-1920x706.jpg'},
  { name: 'Startup Depot', city: 'Lviv', location: '', rating: '★★★★☆', image: 'https://coworking.lviv.ua/wp-content/uploads/2024/08/dsc_5334.jpeg'},
  { name: 'Futura Hub', city: 'Lviv', location: '', rating: '★★★★☆', image: 'https://www.lvivconvention.com.ua/wp-content/uploads/2021/03/kholl-futury.jpg'},
  { name: 'Impact Hub Odessa', city: 'Odessa', location: '', rating: '★★★★☆', image: 'https://impacthub.odessa.ua/wp-content/themes/impact/img/lvng-room.png'},
  { name: '4City', city: 'Odessa', location: '', rating: '★★★★★', image: 'https://renty.ua/imgs/97c69c97faad50bf429d4bd2a549de624b90f9d5906c29889d9a3ef18cf3ffa1'},
  { name: 'Workit', city: 'Odessa', location: '', rating: '★★★★☆', image: 'https://www.coworkbooking.com/images/3711252/kapacita/19175/20218626_115555.jpg'},
  { name: 'Spalah Kharkiv', city: 'Kharkiv', location: '', rating: '★★★★★', image: 'https://lh6.googleusercontent.com/proxy/AnP8ingU4FYj0H3X8ESIVNIIfcgWyp0n75y81d4qJA8Loqk'},
  { name: 'Smart Space', city: 'Kharkiv', location: '', rating: '★★★★★', image: 'https://cdn.pragmatika.media/2025/02/Cluster-Space-07-02-2025-0-29-title.jpg'},
  { name: 'Fabrika.space', city: 'Kharkiv', location: '', rating: '★★★★☆', image: 'https://fabrika.space/wp-content/uploads/2016/02/site-IMG_0018-2.jpg'}
];

if (recommendedCity) {
  const recommendedSpaces = spaces.filter(space => space.city.toLowerCase() === recommendedCity.toLowerCase());

  recommendedSpaces.forEach(space => {
    const spaceCard = document.createElement('div');
    spaceCard.classList.add('space-card');
  });
}

```

Рисунок А.1 – Фрагменти лістингу рекомендаційної системи

Джерело: розроблено автором

```

const app = initializeApp(firebaseConfig);
const auth = getAuth(app);
const db = getDatabase(app);

// Кнопка бронювання
const reserveButton = document.getElementById('reserveButton');

if (reserveButton) {
  reserveButton.addEventListener('click', () => {
    const user = auth.currentUser;
    if (!user) return alert("Увійдіть у свій акаунт");

    const bookingData = {
      location: "Freedom Hub",
      date: "2025-04-14",
      time: "10:00",
      bookedAt: new Date().toISOString()
    };

    const bookingRef = ref(db, `bookings/${user.uid}`);
    push(bookingRef, bookingData)
      .then(() => alert("✅ Заброньовано!"))
      .catch((err) => alert("❌ Помилка: " + err.message));
  });
}

```

Рисунок А.2 – Фрагмент лістингу бронювання через Firebase

Джерело: розроблено автором

```

1  export function setupModal() {
2      const modal = document.getElementById('modal');
3      const closeBtn = modal?.querySelector('.modal-close');
4
5      if (!modal || !closeBtn) return console.error('Не знайдено елементів для модального вікна');
6
7      closeBtn.onclick = () => (modal.style.display = 'none');
8
9      // Клік поза модальним вікном закриває його
10     window.onclick = (e) => {
11         if (e.target === modal) {
12             modal.style.display = 'none';
13         }
14     };
15 }
16
17 export function openModal(data) {
18     const getEl = (id) => document.getElementById(id);
19     const modal = getEl('modal');
20
21     const elements = {
22         title: getEl('modalTitle'),
23         desc: getEl('modalDesc'),
24         city: getEl('modalCity'),
25         image: getEl('modalImage'),
26         price: getEl('modalPrice'),
27         features: getEl('modalFeatures'),
28         map: getEl('modalMap'),
29         bookBtn: getEl('reserveButton'),
30         closeBtn: modal?.querySelector('.modal-close'),
31     };
32
33     // Перевірка наявності всіх елементів
34     if (Object.values(elements).some((el) => !el) || !modal) {
35         return console.error('Не знайдені необхідні елементи модального вікна');
36     }

```

Рисунок А.3 – Фрагмент лістингу модальних вікон

Джерело: розроблено автором

```

<div class="filter-container">
  <label for="cityFilter">Місто:</label>
  <select id="cityFilter">
    <option value="all">Усі міста</option>
    <option value="kyiv">Київ</option>
    <option value="lviv">Львів</option>
    <option value="odesa">Одеса</option>
    <option value="kharkiv">Харків</option>
  </select>
  <label for="featureFilter" style="margin-left: 2rem;">Зручності:</label>
  <select id="featureFilter">
    <option value="all">Усі зручності</option>
    <option value="Wi-Fi">Wi-Fi</option>
    <option value="кухня">Кухня</option>
    <option value="переговорки">Переговорки</option>
    <option value="івенти">Події</option>
  </select>
</div>

```

Рисунок А.4 – Фрагмент лістингу фільтрації

Джерело: розроблено автором