

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
БРОНЮВАННЯ МІСЦЬ У КОВОРКІНГУ

SOFTWARE DEVELOPMENT FOR BOOKING SEATS IN A
COWORKING

Виконав(ла): студент(ка) 4 курсу, групи КНТ-211сп

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

НАДТОЧИЙ В.Е.

(ПРІЗВИЩЕ та ініціали)

Керівник СКРУПСЬКИЙ С.Ю.

(ПРІЗВИЩЕ та ініціали)

Рецензент ГОЛУБ Т.В.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

НАДТОЧОГО Владислава Едуардовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення бронювання місць у коворкінгу.

Software Development for Booking Seats in a Coworking

керівник проєкту (роботи) к.т.н., доцент, СКРУПСЬКИЙ Степан Юрійович

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 24 ” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Теоретичні основи процесу бронювання місць у коворкінгах. 2. Проєктування системи бронювання місць в коворкінгах 3. Опис програмного забезпечення. 4. Тестування програмного забезпечення.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	СКРУПСЬКИЙ С.Ю., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 15 ” квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис)

Владислав НАДТОЧИЙ

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

Степан СКРУПСЬКИЙ

(ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 83 с., 23 табл., 31 рис., 2 дод., 14 джерел.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, БРОНЮВАННЯ, КОВОРКІНГ, ВЕБДОДАТОК, АРХІТЕКТУРА, БАЗА ДАНИХ, ТЕСТУВАННЯ.

Об'єкт дослідження – програмне забезпечення для бронювання місць у коворкінгах.

Предмет дослідження – процеси бронювання місць у коворкінгах та оптимізація взаємодії між користувачами та власниками просторів.

Мета роботи – розробка програмного забезпечення для бронювання місць у коворкінгах, яке дозволить користувачам легко знаходити та бронювати робочі місця, а власникам коворкінгів – ефективно керувати завантаженистю приміщень.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, вебтехнології (HTML, CSS, JavaScript, PHP), системи управління базами даних, методи тестування програмного забезпечення.

Результати. Розроблено програмне забезпечення для бронювання місць у коворкінгах, яке включає вебдодаток з інтуїтивно зрозумілим інтерфейсом користувача, базу даних для зберігання інформації, а також адміністративну панель для управління просторами та взаємодії з клієнтами.

Висновки. Розроблене програмне забезпечення відповідає поставленим вимогам, забезпечує зручність та ефективність процесу бронювання місць у коворкінгах для користувачів.

Галузь використання – управління коворкінгами, надання послуг оренди робочих місць, розвиток стартап-екосистеми та віддаленої роботи.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 83 pages, 23 tables, 31 figures, 2 appendices, 14 sources.

SOFTWARE, BOOKING, COWORKING, WEB APPLICATION, ARCHITECTURE, DATABASE, TESTING.

Object of research - software for booking seats in coworking spaces.

Subject of research - processes of booking seats in coworking spaces and optimization of interaction between users and space owners.

The purpose of the work is to develop software for booking seats in coworking spaces, which will allow users to easily find and book workplaces, and coworking owners to effectively manage the occupancy of the premises.

Materials, methods, and technical means: structural and object-oriented programming, web technologies (HTML, CSS, JavaScript, PHP), database management systems, software testing methods.

Results. Developed software for booking seats in coworking spaces, which includes a web application with an intuitive user interface, a database for storing information, as well as an admin panel for managing spaces and interacting with clients.

Conclusions. The developed software meets the specified requirements, provides convenience and efficiency of the booking process for users in coworking spaces.

Field of use - management of coworking spaces, provision of workplace rental services, development of startup ecosystem and remote work.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Теоретичні основи процесу бронювання місць у коворкінгах.....	10
1.1 Огляд сучасного стану ринку послуг коворкінгів.....	10
1.2 Аналіз існуючих систем бронювання місць у коворкінгах.....	11
1.3 Вимоги до розробки програмного забезпечення для бронювання місць у коворкінгах.....	16
1.4 Висновки за розділом.....	19
2 Проєктування системи бронювання місць у коворкінгах.....	21
2.1 Аналіз вимог до системи.....	21
2.2 Проєктування архітектури програмного забезпечення.....	24
2.3 Вибір технологій та інструментів для реалізації.....	28
2.4 Висновки за розділом.....	32
3 Опис програмного забезпечення.....	34
3.1 База даних.....	34
3.2 Опис структури вебсайту.....	38
3.3 Інтерфейс користувача.....	46
3.4 Висновки за розділом.....	53
4 Тестування програмного забезпечення.....	55
4.1 Модульне тестування.....	55
4.2 Тестування користувацького інтерфейсу.....	58
4.3 Висновки за розділом.....	59
Висновки.....	61
Перелік джерел посилання.....	63
Додаток А Текст програми.....	64
Додаток Б Слайди презентації.....	76

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	- Application Programming Interface;
CSS	- Cascading Style Sheets;
DOM	- Document Object Model;
HTML	- HyperText Markup Language;
JSON	- JavaScript Object Notation;
NoSQL	- Not only SQL;
REST	- Representational State Transfer;
SPA	- Single Page Application;
SQL	- Structured Query Language;
UI	- User Interface (інтерфейс користувача);
IT	- інформаційні технології.

ВСТУП

У сучасному світі, де віддалена робота та фріланс стають усе більш поширеними, коворкінги відіграють важливу роль у забезпеченні гнучких та ефективних робочих просторів. Розвиток цифрових технологій відкриває нові можливості для оптимізації процесів бронювання та управління місцями у коворкінгах. Розробка програмного забезпечення для бронювання місць у коворкінгах є актуальною та перспективною задачею, яка потребує глибокого аналізу потреб користувачів та використання сучасних ІТ-рішень.

З поширенням віддаленої роботи та розвитком стартап-екосистеми попит на коворкінги в Україні зростає. Автоматизація процесів бронювання робочих місць та управління простором стає необхідністю для ефективної роботи коворкінгів. Впровадження веб та мобільних додатків для бронювання місць дозволяє користувачам легко знаходити та резервувати робочі місця, а власникам коворкінгів - оптимізувати завантаженість приміщень [1].

Інтеграція з календарями, системами онлайн-оплати та управління доступом стає важливою складовою програмного забезпечення для коворкінгів. Використання аналітики даних допомагає власникам коворкінгів краще розуміти потреби користувачів та оптимізувати роботу просторів [2].

Метою даного дипломного проекту є розробка програмного забезпечення для бронювання місць у коворкінгах, яке дозволить користувачам легко знаходити та бронювати робочі місця, а власникам коворкінгів - ефективно керувати завантаженістю приміщень.

Предметом дослідження є процеси бронювання місць у коворкінгах та оптимізація взаємодії між користувачами та власниками просторів. Об'єктом дослідження є програмне забезпечення, що включає вебсайт та мобільний додаток для бронювання місць у коворкінгах.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- аналіз потреб користувачів та вимог до функціоналу системи бронювання місць у коворкінгах;
- дослідження сучасних ІТ-рішень для управління коворкінгами та бронювання місць;
- проєктування архітектури програмного забезпечення, що забезпечує масштабованість, безпеку та зручність використання;
- розробка бази даних для зберігання інформації про коворкінги, користувачів та бронювання;
- реалізація функціоналу бронювання місць, управління коворкінгами та оплати послуг;
- тестування розробленого програмного забезпечення.

Виконання цих завдань дозволить створити ефективне програмне забезпечення для бронювання місць у коворкінгах, яке відповідатиме потребам користувачів та власників просторів, сприяючи розвитку ринку коворкінгів в Україні.

1 ТЕОРЕТИЧНІ ОСНОВИ ПРОЦЕСУ БРОНЮВАННЯ МІСЦЬ У КОВОРКІНГАХ

1.1 Огляд сучасного стану ринку послуг коворкінгів

Ринок послуг коворкінгів в Україні активно розвивається, відповідаючи на зростаючий попит з боку фрілансерів, стартапів та малих підприємств. Коворкінги пропонують гнучкі робочі простори без довгострокових зобов'язань, що є привабливим для багатьох професіоналів. Ринок характеризується різноманітністю форматів коворкінгів, від простих офісних приміщень до креативних просторів з додатковими послугами [3].

Основні тенденції та особливості ринку коворкінгів в Україні:

- зростання кількості коворкінгів у великих містах, таких як Київ, Львів, Одеса та Харків, де концентрація фрілансерів та стартапів найвища;
- фокус на ергономіці робочих місць та наявності зон відпочинку для забезпечення комфорту та продуктивності користувачів;
- розвиток спеціалізованих коворкінгів, орієнтованих на певні професійні спільноти, наприклад, IT-фахівців, дизайнерів, маркетологів тощо;
- інтеграція додаткових сервісів, таких як конференц-зали, зони для нетворкінгу, кафе, що створюють додаткову цінність для користувачів;
- використання цифрових технологій для автоматизації процесів бронювання місць, управління доступом та оплати послуг.

Незважаючи на активний розвиток, ринок коворкінгів в Україні стикається з певними викликами (рис. 1.1.). Висока конкуренція у великих містах змушує коворкінги шукати способи диференціації та покращення якості послуг. Пандемія COVID-19 також вплинула на ринок, змусивши коворкінги адаптуватися до нових умов роботи, вживати заходів безпеки та пропонувати гнучкі умови оренди.

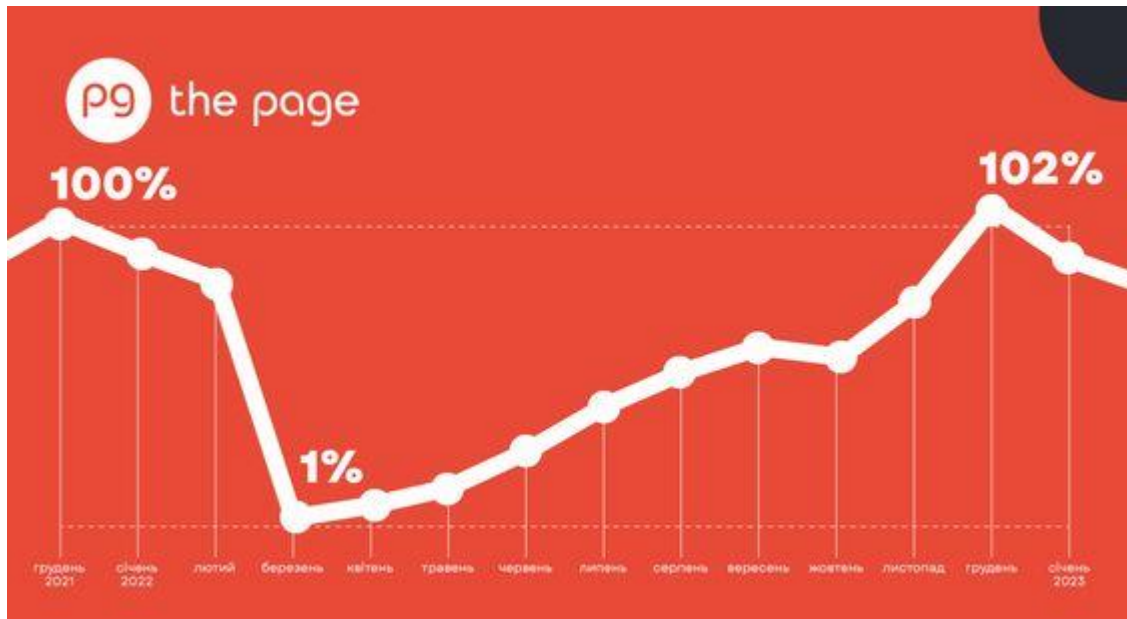


Рисунок 1.1 - Динаміка бронювань коворкінгів в Україні [4]

Перспективи розвитку ринку коворкінгів в Україні залишаються оптимістичними. Зростання популярності віддаленої роботи та фрілансу, розвиток стартап-екосистеми та потреба у гнучких робочих просторах сприяють попити на послуги коворкінгів. Використання сучасних технологій, зокрема, програмного забезпечення для бронювання місць та управління коворкінгами, відкриває нові можливості для оптимізації роботи просторів та покращення користувацького досвіду.

1.2 Аналіз існуючих систем бронювання місць у коворкінгах

Для розуміння потреб ринку та визначення вимог до розроблюваного програмного забезпечення важливо проаналізувати існуючі системи бронювання місць у коворкінгах. На основі дослідження доступних рішень можна виділити основні функції та особливості, які користувачі очікують від подібних систем [5].

Розглянемо декілька прикладів існуючих систем бронювання місць у коворкінгах.

На рисунку 1.2 відображено вебінтерфейс для бронювання на сайті коворкінгу IC Coworking. Інтерфейс містить кілька основних елементів та функцій. У верхній частині сторінки розміщено меню з основними розділами сайту, такими як "Робочі місця", "Офіси", "Переговорні кімнати" та "Конференц-зали". Також доступні розділи "Про нас", "Тарифи", "News Блог" і мовні налаштування (українська, англійська). Праворуч знаходиться кнопка для бронювання [6].

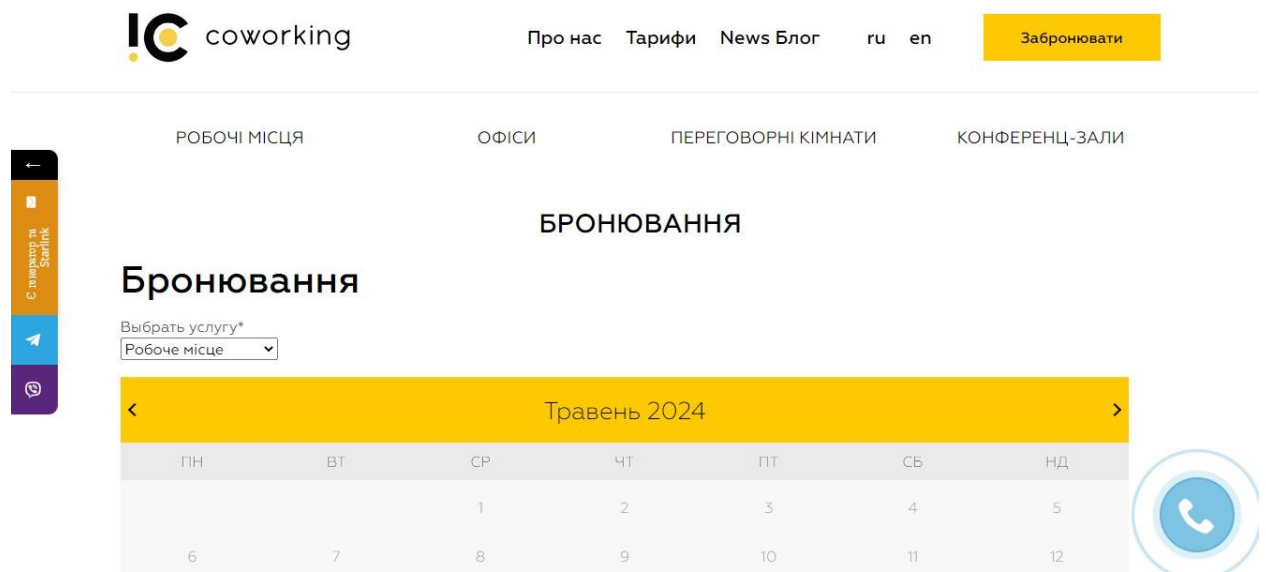


Рисунок 1.2 – IC Coworking [7]

На сторінці бронювання користувач може вибрати послугу з випадаючого меню, яке наразі показує опцію "Робоче місце". Під цим меню розташований календар на травень 2024 року, де можна вибрати дати для бронювання. Інтерфейс також містить кнопки для переміщення по місяцях календаря. В правому нижньому куті є іконка телефону для зв'язку з адміністрацією або службою підтримки.

Сайт IC Coworking надає можливість користувачам швидко та зручно бронювати робочі місця, офіси, переговорні кімнати та конференц-зали. Інтуїтивний дизайн і зрозумілі опції навігації роблять процес бронювання простим і ефективним.

На рисунку 1.3 представлено головну сторінку вебсайту ХАБковоркінг, яка орієнтована на простоту та зручність у використанні. Основний фокус зосереджений на промоції коворкінгу з чітко виділеним закликом до дії, що пропонує відвідувачам спробувати простір безкоштовно. Заголовок "КОВОРКІНГ" розміщений на фоні відтінків червоного з адресою на вулиці Героїв УПА, 80, що може сприяти легкій ідентифікації розташування.

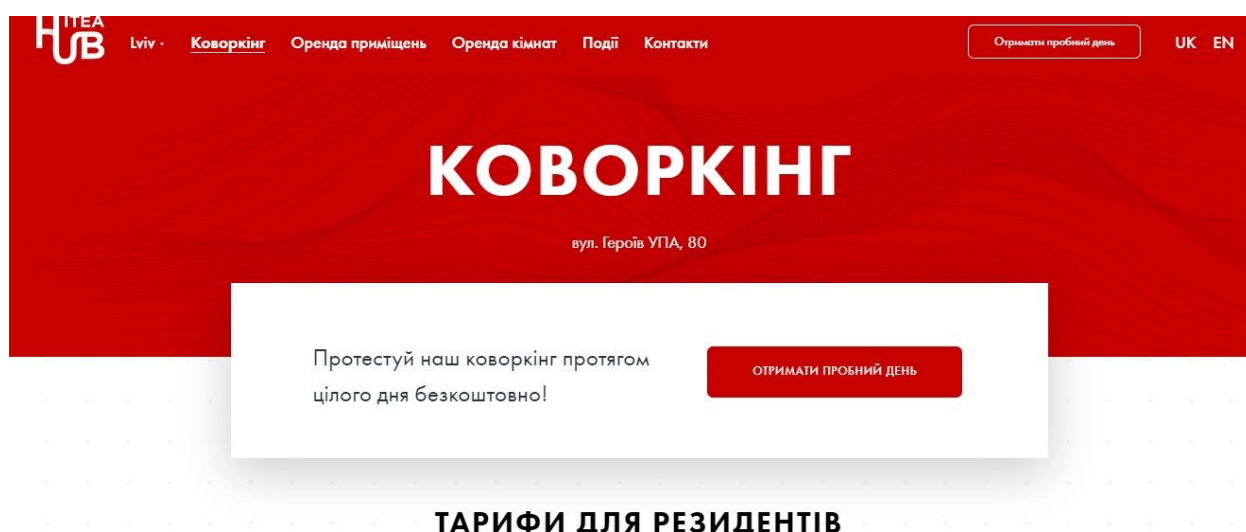


Рисунок 1.3 – ХАБковоркінг [2]

Внизу сторінки видно розділ "Тарифи для резидентів", що натякає на доступність детальної інформації про вартість послуг. Навігаційна панель у верхній частині сторінки містить кілька вкладок, включаючи "Користь", "Огляд приміщень", "Оренда кімнат", "Події" та "Контакти", дозволяючи користувачам легко переходити до інформації, яка їх цікавить. Можливість перемикання між українською та англійською версіями сайту говорить про міжнародну орієнтацію ХАБковоркінгу та його готовність обслуговувати різноманітну аудиторію [3].

На рисунку 1.4 представлено головну сторінку вебсайту "РуумПростір", який здається сучасним та стильним, з акцентом на візуальну привабливість та зручність використання. Головний візуальний елемент — це

назва "РуумПростір", виконана у креативному дизайні з використанням різних кольорів та текстур, що символізує креативність і інноваційність простору. Праворуч розміщено два зображення: одне показує засідання у світлій кімнаті з людьми, що активно обговорюють питання, а інше — більш неформальне місце для відпочинку з м'якими мішками для сидіння [4].

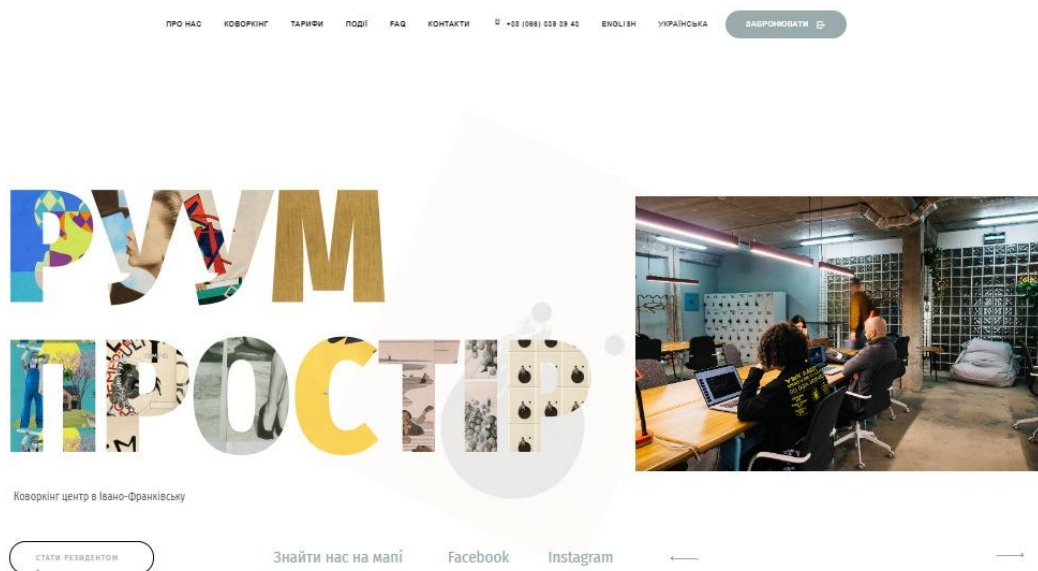


Рисунок 1.4 – РуумПростір

У верхній частині сторінки розташована навігаційна панель із розділами "Про нас", "Програми", "Спільнота", "Блог", "Контакти", що дозволяє користувачам швидко знайти потрібну інформацію про організацію, її діяльність і можливості для співпраці. Також є кнопки для швидкого доступу до облікового запису користувача та реєстрації/входу, що забезпечує зручність взаємодії з сайтом [8].

Внизу головної сторінки знаходяться посилання на сторінки "РуумПростір" у соціальних мережах, таких як Facebook та Instagram, що сприяє залученню спільноти та підтриманню зв'язку з відвідувачами. Все це створює враження про дружній до користувача сайт, який підтримує активну спілкування та співпрацю в рамках коворкінгового простору.

Порівняльна таблиця 1.1 для трьох описаних вебсайтів коворкінгів.

Таблиця 1.1 – Порівняння сучасних вебсайтів

Характеристика	IS Coworking	ХАБковоркінг	РуумПростір
Основний фокус	Бронювання робочих місць	Інформація про коворкінг та тарифи	Презентація коворкінгу та його спільноти
Дизайн	Мінімалістичний із зручним календарем	Стильний із яскравими візуальними акцентами	Сучасний, креативний дизайн з візуальною привабливістю
Функції	Вибір послуг і дати бронювання	Інформація про приміщення, події, контакти	Інформація про програми, спільноту, блог
Мовні опції	Українська, російська, англійська	Українська, англійська	Українська
Інтерактивні елементи	Кнопка бронювання, календар	Заклик до дії для безкоштовної пробної візиту	Посилання на соціальні мережі, реєстрація/вхід
Соціальні мережі	Контакти через телефон	Посилання на Facebook, Instagram	Посилання на Facebook, Instagram

Ця таблиця допомагає порівняти основні характеристики кожного з сайтів, їхній дизайн, функціонал і зручності для користувача, а також як вони презентують свої послуги і залучають аудиторію.

Аналізуючи ці та інші існуючі системи бронювання місць у коворкінгах, можна виділити наступні ключові особливості та функції:

- можливість бронювання робочих місць, конференц-залів та інших приміщень у реальному часі;
- наявність календаря для перегляду доступності приміщень;
- можливість керування бронюваннями (змінення, скасування);
- інтеграція з системами управління членством та білінгом;
- наявність мобільних додатків для зручності користувачі;
- аналітика та звітність для власників коворкінгів.

Врахування цих особливостей та функцій при розробці власного програмного забезпечення для бронювання місць у коворкінгах дозволить створити продукт, що відповідає очікуванням користувачів та потребам ринку.

1.3 Вимоги до розробки програмного забезпечення для бронювання місць у коворкінгах

На основі аналізу ринку та існуючих рішень для бронювання місць у коворкінгах, можна сформулювати ключові вимоги до розробки власного програмного забезпечення [9]. Ці вимоги охоплюють функціональні та нефункціональні аспекти системи, які необхідно врахувати при проектуванні та реалізації продукту.

Функціональні вимоги:

- реєстрація та автентифікація користувачів: система повинна надавати можливість реєстрації нових користувачів та безпечної автентифікації існуючих користувачів;
- бронювання місць: користувачі повинні мати можливість легко бронювати робочі місця, конференц-зали та інші приміщення коворкінгу на певний період часу;

- календар доступності: система повинна відображати доступність приміщень у реальному часі за допомогою календаря, що дозволить користувачам бачити вільні та заброньовані місця;
- керування бронюванням: користувачі повинні мати можливість переглядати, змінювати та скасовувати свої бронювання;
- платежі: система повинна підтримувати інтеграцію з платіжними шлюзами для обробки оплати за бронювання місць;
- управління коворкінгами: власники коворкінгів повинні мати можливість керувати інформацією про свої простори, додавати та редагувати приміщення, встановлювати ціни та правила бронювання;
- сповіщення: система повинна надсилати користувачам сповіщення про підтвердження бронювань, зміни в розкладі та інші важливі події.

Нефункціональні вимоги:

- безпека: система повинна забезпечувати високий рівень безпеки даних користувачів та транзакцій, використовуючи шифрування та інші заходи безпеки;
- продуктивність: система повинна витримувати високе навантаження та забезпечувати швидкий відгук для користувачів;
- масштабованість: архітектура системи повинна дозволяти легке масштабування для підтримки зростаючої кількості користувачів та коворкінгів;
- зручність використання: інтерфейс користувача повинен бути інтуїтивно зрозумілим та простим у навігації, забезпечуючи позитивний досвід користувачів;
- крос-платформеність: система повинна бути доступною через веббраузери та мобільні пристрої (iOS та Android) для зручності користувачів;
- надійність: система повинна забезпечувати стабільність роботи та мінімізувати час простою.

Таблиця 1.3 - Огляд ключових завдань

Етап	Опис завдань
Визначення потреб користувачів	Аналіз потреб кінцевих користувачів та адміністрації коворкінгів для визначення функціональних вимог до системи.
Аналіз конкурентних рішень	Дослідження існуючих систем бронювання для визначення їх сильних та слабких сторін.
Формулювання вимог до системи	Формулювання технічних та функціональних вимог до нової системи на основі зібраної інформації.
Вибір технологічного стеку	Вибір мов програмування, фреймворків, баз даних та інших технологій для реалізації системи.
Проектування архітектури системи	Створення архітектурної схеми системи, яка описує взаємодію компонентів, модульну структуру та ін.
Розробка інтерфейсу користувача	Дизайн інтерфейсу користувача, який має бути інтуїтивно зрозумілим та зручним.
Розробка основних модулів	Реалізація ключових модулів системи, включаючи модуль бронювання, управління закладом та облік клієнтів.
Тестування	Перевірка системи на наявність помилок, відповідність вимогам і надійність роботи.
Впровадження та налагодження	Запуск системи в експлуатацію з подальшим моніторингом і налагодженням її роботи.
Підтримка та розвиток	Забезпечення оперативної технічної підтримки, оновлення та розширення функціоналу системи.

Врахування цих вимог при розробці програмного забезпечення для бронювання місць у коворкінгах дозволить створити продукт, що відповідає потребам користувачів та власників коворкінгів, забезпечуючи ефективне управління робочими просторами та зручність використання.

1.4 Висновки за розділом

У першому розділі було розглянуто теоретичні основи процесу бронювання місць у коворкінгах та проаналізовано сучасний стан ринку послуг коворкінгів в Україні.

Ринок коворкінгів в Україні активно розвивається, відповідаючи на зростаючий попит з боку фрілансерів, стартапів та малих підприємств. Основні тенденції включають збільшення кількості коворкінгів у великих містах, фокус на ергономіці та додаткових послугах, розвиток спеціалізованих просторів та використання цифрових технологій для автоматизації процесів.

Аналіз існуючих систем бронювання місць у коворкінгах, таких як IC Coworking, ХАБковоркінг, РуумПростір, дозволив виділити ключові особливості та функції, які користувачі очікують від подібних систем. Серед них можливість бронювання в реальному часі, наявність календаря доступності, керування бронюваннями, інтеграція з системами управління членством та білінгом, наявність мобільних додатків та аналітика для власників коворкінгів.

На основі аналізу ринку та існуючих рішень було сформульовано функціональні та нефункціональні вимоги до розробки власного програмного забезпечення для бронювання місць у коворкінгах. Функціональні вимоги включають реєстрацію та автентифікацію користувачів, бронювання місць, календар доступності, керування бронюваннями, платежі, управління коворкінгами та сповіщення. Нефункціональні вимоги охоплюють безпеку,

продуктивність, масштабованість, зручність використання, крос-платформеність та надійність.

Врахування цих вимог при розробці програмного забезпечення для бронювання місць у коворкінгах дозволить створити продукт, що відповідає потребам ринку та очікуванням користувачів, забезпечуючи ефективне управління робочими просторами та зручність використання.

У наступному розділі буде розглянуто процес проєктування архітектури програмного забезпечення, вибір технологій та інструментів для реалізації, а також розробку ключових компонентів системи.

2 ПРОЄКТУВАННЯ СИСТЕМИ БРОНЮВАННЯ МІСЦЬ У КОВОРКІНГАХ

У цьому розділі буде розглянуто процес проєктування архітектури програмного забезпечення для системи бронювання місць у коворкінгах. Буде проведено аналіз вимог до системи, розроблено архітектуру та обрано технології й інструменти для реалізації.

2.1 Аналіз вимог до системи

Функціональні вимоги визначають як система повинна виконувати свої завдання, включно з обробкою, обміном та керуванням даними для бронювання місць у коворкінгу. Через різноманітність користувацьких ролей у системі, функціональні вимоги для кожної категорії користувачів визначаються окремо. Для клієнтів функціональні вимоги включають (табл. 2.1):

- вхід в систему через електронну пошту та пароль;
- процес реєстрації, який вимагає вказання імені, прізвища, унікальної електронної адреси, номера телефону та створення пароля;
- можливість незареєстрованим користувачам переглядати список коворкінгів та інформацію про кожен на сторінці без можливості переглядати вільні місця чи робити бронювання;
- детальна інформація на сторінці коворкінгу, включаючи назву, зображення, адресу, відгуки користувачів та доступні послуги;
- функції сортування та фільтрації коворкінгів за ціною, рейтингом, популярністю та наявністю місць на певну дату;
- пошук коворкінгів за назвою або локацією та перегляд схожих коворкінгів;
- бронювання місця з можливістю вибору дати, часу та специфічного місця;

- можливість зміни або скасування бронювань та перегляд історії бронювань;
- функції оцінювання коворкінгу та залишення відгуку після відвідування.

Таблиця 2.1 – Функції для користувача-клієнта

№	Назва функції	Скорочений опис функції
1	Вхід у систему	Авторизація через електронну пошту та пароль.
2	Реєстрація	Введення особистих даних для створення акаунту.
3	Перегляд коворкінгів	Перегляд інформації про коворкінги без можливості бронювання.
4	Деталі коворкінгу	Показує повну інформацію про коворкінг.
5	Сортування та фільтрація коворкінгів	Фільтрування і сортування за ціною, рейтингом, доступністю місць.
6	Пошук та перегляд схожих коворкінгів	Шукати коворкінги за назвою чи локацією.
7	Бронювання місць	Резервування місця за конкретні дату і час.
8	Управління бронюваннями	Можливість редагувати чи скасовувати бронювання.
9	Оцінювання та відгуки	Залишення відгуків та оцінок після відвідування.

Функціональні вимоги для адміністратора коворкінгу включають створення та редагування інформації про коворкінг, приховування коворкінгу у разі його закриття, перегляд всіх бронювань, асоційованих з

їхнім коворкінгом, і створення нових бронювань для обліку всіх резервацій (табл. 2.2.).

Таблиця 2.2 - Функції для адміністратора

№	Назва функції	Опис функції
1	Управління інформацією про коворкінг	Додавання, оновлення чи видалення інформації про коворкінги.
2	Перегляд бронювань	Доступ до всіх бронювань коворкінгу для ефективного управління.
3	Створення бронювань персоналом	Внесення в систему бронювань, здійснених через інші канали.
4	Реєстрація нових коворкінгів	Створення облікових записів для нових коворкінгів.

Нефункціональні вимоги описують якісні характеристики програмного забезпечення, такі як ефективність, легкість у використанні, стабільність і час відгуку, які зазначені в SRP-документі. Ці вимоги не визначають конкретні функції системи, але встановлюють якість їх виконання. Вони є обмеженнями або стандартами, що впливають на робочий процес розробників та користувачів. Оцінка нефункціональних вимог часто є суб'єктивною і вимагає оцінювальних тестів, а не прямих тестувань.

У системі бронювання місць у коворкінгу визначено, що кожен користувач може створити тільки один аккаунт, прикріплений до унікальної електронної адреси та номера телефону. Паролі, які використовуються для доступу до системи, повинні складатися мінімум з шести символів і зберігаються в зашифрованому форматі у базі даних. Система має надавати користувачам повідомлення про помилки у разі введення некоректних даних під час реєстрації чи входу.

Неавторизовані користувачі можуть переглядати список коворкінгів, але не можуть бронювати місця до моменту авторизації. Авторизовані

користувачі здатні доступатися тільки до власних бронювань. Дати бронювань не можуть бути задані на минуле, і час завершення завжди має перевищувати час початку [2].

Вибір місць у коворкінгу обмежується тільки тими, які є вільними на момент бронювання, і перевірка наявності місць можлива тільки для окремо вибраного коворкінгу. Адміністратори мають змогу редагувати або приховувати інформацію тільки про ті коворкінги, якими вони керують, що забезпечує додатковий контроль та управління відповідно до їх облікових записів.

2.2 Проєктування архітектури програмного забезпечення

Процес проєктування архітектури програмного забезпечення є ключовим етапом розробки, що вимагає ретельного аналізу та стратегічного планування. Архітектурне проєктування включає визначення структури системи, її компонентів, їх взаємодії між собою та зовнішнім середовищем, а також технологічних рішень, які забезпечать необхідну функціональність, продуктивність, масштабованість та безпеку (табл. 2.3).

Таблиця 2.3 - Проєктування архітектури програмного забезпечення

Аспект	Опис
Модульність і зв'язність	Архітектура має бути розділена на незалежні модулі, як-от UI, управління БД, обробка бронювань, для спрощення інтеграції та підтримки.
Клієнт-серверна архітектура	Використання централізованої обробки даних для забезпечення стабільного сервісу через різні інтерфейси, включаючи веб і мобільні додатки.
Висока доступність та масштабованість	Система повинна легко масштабуватися для обробки збільшеного навантаження, з використанням хмарних рішень для додаткової гнучкості.

Продовження таблиця 2.3

Аспект	Опис
Безпека	Фокус на захисті даних користувачів, захисті від зовнішніх загроз і забезпеченні цілісності даних, відповідно до найкращих практик і стандартів.
Інтерфейси та інтеграція	Розробка гнучких інтерфейсів для легкої інтеграції з іншими системами і сервісами, що дозволяє розширити функціонал та інтегрувати систему у ширшу екосистему.
Тестування та деплоймент	Розробка чіткої стратегії тестування та розгортання з метою забезпечення легкості оновлень системи і мінімізації часу її недоступності.

На рисунку 2.1 представлено схему відносин між різними елементами системи. В центрі розташована "Система", яка має "Архітектуру". "Архітектура" складається з "Опису архітектури" та пов'язана з "Моделлю" через "Подання", яке є одиницею відношення між "Описом архітектури" та "Моделлю". "Опис архітектури", у свою чергу, пов'язаний з "Точкою зору", яка містить інтереси "Учасника". "Учасник" може мати кілька інтересів, що розглядаються через різні "Точки зору".

Така діаграма використовується для відображення структури системи з точки зору її архітектури, ролей, які відіграють учасники, і способів представлення архітектури. Взаємозв'язки з "n" та "m" позначають кількість зв'язків між елементами, де "n" може означати багато, а "m" - один або більше. Наприклад, одна "Точка зору" може відображати кілька "Подань", які, у свою чергу, пов'язані з "Моделлю" [4].

Це типова схема для візуалізації відносин в рамках архітектурних рамок, таких як ISO/IEC/IEEE 42010, що описує рекомендовану практику для опису архітектури систем в умовах стейкхолдерів та їхніх інтересів.

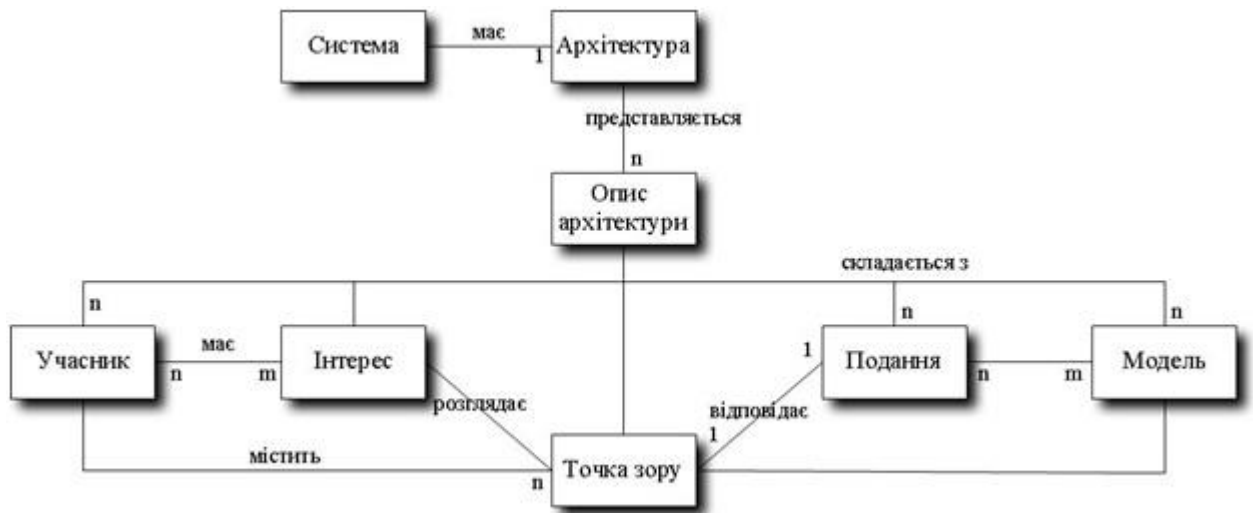


Рисунок 2.1 – Мета модель архітектури

Основна мета проектування архітектури програмного продукту полягає у зниженні витрат часу та зусиль на його створення та обслуговування. Таке програмне забезпечення повинно бути гнучким, здатним до швидкої адаптації до змін. Рівень складності модифікацій має відповідати розміру необхідних змін, а не бути ускладненим через їхню природу.

Архітектура існує для підтримки повного життєвого циклу програмного забезпечення, сприяючи його зручності у використанні, ефективності розробки, обслуговування та деплойменту. Її вища мета — мінімізація загальних витрат протягом усього життя системи та підвищення продуктивності розробників.

У рамках даного проєкту передбачається виразне розмежування між бекендом і фронтом компонентів. Структура вебдодатку буде відповідати організації, яка представлена на рисунку 2.2.

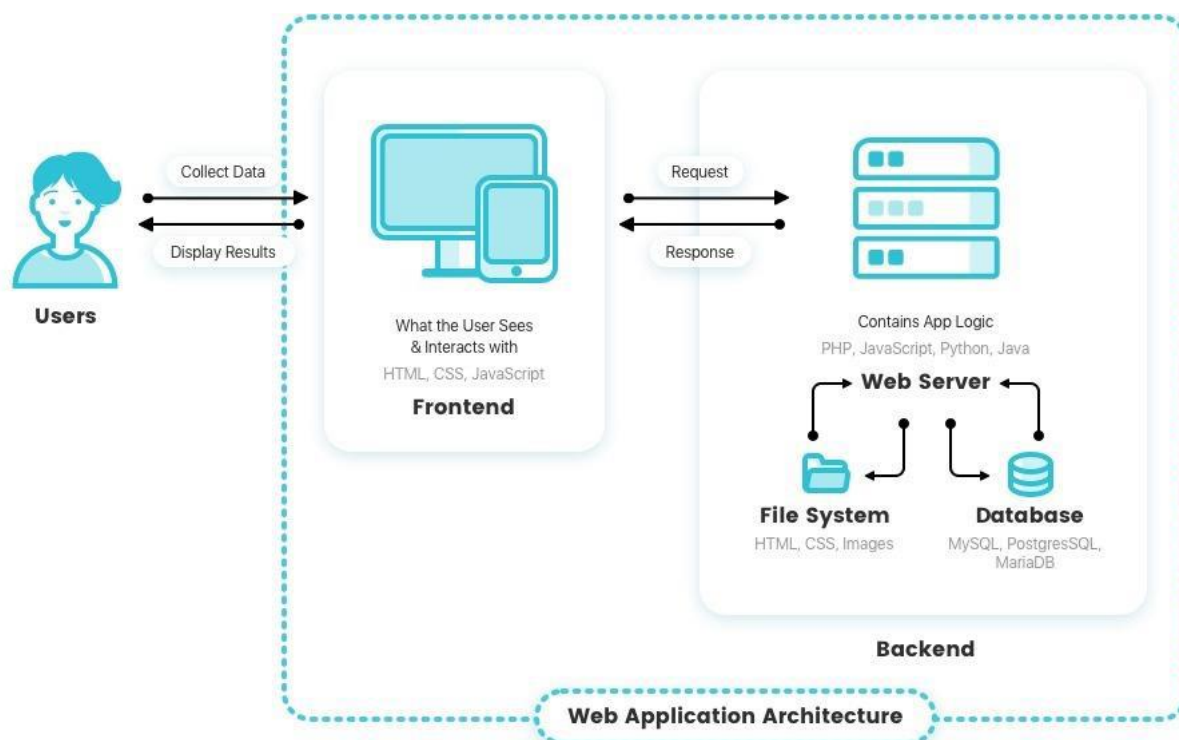


Рисунок 2.2 – Архітектура вебдодатку

На зображенні представлена схема архітектури вебдодатку, яка складається з двох основних компонентів: фронтенду та бекенду.

Фронтенд:

- це та частина вебдодатку, з якою безпосередньо взаємодіють користувачі;
- фронтенд збирає дані від користувачів і відображає результати їхніх запитів;
- основні технології, використовувані на фронтенді, включають HTML, CSS та JavaScript, що дозволяють створювати структуру, стиль і інтерактивність вебсторінок.

Бекенд:

- бекенд містить логіку додатку, яка обробляє запити від фронтенду і відправляє відповіді;
- бекенд складається з вебсервера, файлової системи і бази даних;
- вебсервер може бути написаний на мовах програмування PHP, JavaScript (Node.js), Python, Java та інших. Він приймає запити від

користувачів через фронтенд, обробляє ці запити відповідно до бізнес-логіки додатку, і взаємодіє з файловою системою та базою даних перед тим, як надсилати відповідь;

- файлова система зберігає файли, такі як HTML, CSS, зображення та скрипти, які необхідні для роботи вебдодатку;
- база даних: зазвичай використовуються системи управління базами даних, такі як MySQL, PostgreSQL або MariaDB, для зберігання, організації, доступу та управління даними.

Вся взаємодія між фронтендом і бекендом відбувається через запити та відповіді, які передаються через інтернет. Це забезпечує динамічність та інтерактивність вебдодатків, дозволяючи користувачам отримувати персоналізований досвід користування.

2.3 Вибір технологій та інструментів для реалізації

Фронтенд вебдодатку, згідно з описаною архітектурою, використовує технології HTML, CSS та JavaScript. Кожна з цих технологій відіграє ключову роль у створенні вебінтерфейсів, що є інтерактивними та зручними для користувача.

HTML (HyperText Markup Language) відповідає за структурний аспект вебсторінки, визначаючи елементи, такі як заголовки, параграфи, списки, посилання та інші структурні компоненти. Це фундамент веброзробки, який дозволяє іншим технологіям впливати на вміст та представлення вебсторінки.

CSS (Cascading Style Sheets) використовується для оформлення вебсторінки. CSS керує візуальним стилем елементів HTML, включаючи кольори, шрифти, відступи, вирівнювання та багато іншого. Завдяки CSS, розробники можуть створювати різні теми та макети для вебдодатків, роблячи інтерфейси привабливішими та функціональнішими.

JavaScript відповідає за інтерактивність вебсторінок. Він дозволяє реалізувати різноманітні функції, такі як відповіді на кліки користувачів, динамічне оновлення вмісту, анімації, обробку даних форм та інші динамічні елементи інтерфейсу. JavaScript також є основою для численних фреймворків і бібліотек, які подальше спрощують розробку вебдодатків (табл. 2.4).

Таблиця 2.4 – Порівняння популярних бібліотек та фреймворків

Назва плагіну	Основна функція	Типове використання	Переваги	Недоліки
jQuery	Маніпуляція DOM, обробка подій	Робота з елементами HTML, події, анімація	Легко використовувати, широка підтримка	Може бути надмірним для сучасних SPA
React.js	Створення інтерфейсів користувача	Розробка односторінкових додатків (SPA)	Декларативний, компонентний підхід	Вимагає навчання, оптимізації стану
Vue.js	Прогресивний JavaScript фреймворк	Розробка інтерактивних вебінтерфейсів	Легкий, інтуїтивно зрозумілий	Спільнота менша, ніж у React
Angular	Повноцінний фреймворк для вебдодатків	Розробка великих, складних вебдодатків	Сильна модель, об'єднаність	Складний у навчанні, великий розмір
Lodash	Утиліти для роботи з масивами, об'єктами і т.д.	Оптимізація коду, спрощення маніпуляцій з даними	Висока продуктивність, гнучкість	Можливість перевантаження з залежностями
D3.js	Візуалізація даних	Створення складних графіків та діаграм	Гнучкість, потужний у візуалізації даних	Висока крива навчання

Ця таблиця дозволяє оцінити плагіни JavaScript з різних аспектів, включаючи їхні основні функції, типові сценарії використання, а також має перелік переваг та можливих недоліків. Вибір конкретного плагіну залежатиме від потреб проєкту, рівня складності додатку та досвіду розробника.

За допомогою цих трьох технологій, фронтенд розробники можуть створювати гнучкі та адаптивні вебінтерфейси, які забезпечують користувачам якісний досвід взаємодії з вебдодатком. Це забезпечує основу для створення ефективних та привабливих вебсторінок, які можуть працювати на різних пристроях та платформах.

Бекенд вебдодатку, згідно з описаною архітектурою, складається з вебсервера, файлової системи та бази даних. Розглянемо кожен компонент більш детально, а також популярні технології, що використовуються для їх реалізації.

Вебсервер відповідає за обробку запитів, що надходять від фронтенду, і забезпечує зворотний зв'язок у вигляді відповідей. Це програмне забезпечення, яке може використовувати різноманітні мови програмування для реалізації логіки додатку. Популярні мови та технології для бекенду включають PHP, JavaScript (за допомогою Node.js), Python і Java. Кожна мова має свої бібліотеки та фреймворки, які спрощують розробку і підвищують продуктивність розробника. Наприклад, PHP використовується з такими фреймворками, як Laravel і Symfony, Node.js часто використовує Express.js, Python відомий за Django та Flask, а Java - за Spring і Hibernate.

Ця таблиця допомагає визначити сильні та слабкі сторони кожного фреймворку або бібліотеки, їхні основні мови програмування, а також типи проєктів, для яких вони найбільш підходящі. Вибір конкретного фреймворку часто залежить від конкретних потреб проєкту, досвіду розробників та інших факторів, таких як вимоги до продуктивності та масштабованості.

Таблиця 2.5 – Порівняння популярних мов та технологій для бекенду

Фреймворк/ Бібліотека	Мова	Переваги	Недоліки	Прикладне використання
Express.js	JavaScript	Легкий, гнучкий, підтримує мідлвари	Менш строгий у структурі, може потребувати додаткового коду	Прості REST API, вебдодатки на Node.js
Django	Python	Високий рівень абстракції, багато вбудованих можливостей	Може бути важковажким для дуже простих сайтів	Великі вебдодатки зі складною бізнес- логікою
Flask	Python	Мінімалістичний, гнучкий	Потребує додаткових розширень для повноцінної функціональності	Легкі API, мікросервіси
Spring Boot	Java	Широкі можливості, велика спільнота, велика кількість плагінів	Велика кількість конфігурації, круте навчання	Корпоративні вебдодатки, мікросервіси
Laravel	PHP	Чистий код, інтегровані засоби тестування	Може бути важким для вивчення для початківців	Вебдодатки, які вимагають швидкої розробки
.NET Core	C#	Підтримка крос- платформеності, висока продуктивність	Залежить від екосистеми Microsoft	Корпоративні рішення, хмарні додатки

Файлова система забезпечує зберігання статичних файлів, таких як HTML, CSS, зображення та скрипти. Це важливо для забезпечення необхідних ресурсів, що використовуються вебдодатком.

Ефективне управління файловою системою важливе для швидкості завантаження вебсторінок та загальної продуктивності вебдодатку.

Використання CDN (Content Delivery Network) може допомогти оптимізувати цей процес, розподіляючи вміст для зменшення затримок та навантаження на сервер.

Бази даних відіграють ключову роль в бекенді, оскільки вони забезпечують зберігання та організацію даних, які використовуються вебдодатком. Бази даних можуть бути реляційними, як MySQL, PostgreSQL та MariaDB, або NoSQL, як MongoDB і Cassandra.

Вибір між реляційними та NoSQL базами даних залежить від вимог до структури даних, масштабованості та швидкості запитів. Реляційні бази даних надають сильні гарантії умовності, що є критично важливим для транзакційних систем. NoSQL бази даних часто вибирають за їх гнучкість та масштабованість при роботі з великими обсягами неструктурованих даних.

Цей аналіз допомагає зрозуміти, як кожен компонент бекенду впливає на загальну роботу вебдодатку та як вибір технологій може вплинути на ефективність розробки та експлуатації.

2.4 Висновки за розділом

У другому розділі було проведено проєктування системи бронювання місць у коворкінгах, що включало аналіз вимог, розробку архітектури та вибір технологій і інструментів для реалізації.

На основі аналізу функціональних та нефункціональних вимог до системи було визначено ключові аспекти, які повинні бути враховані при розробці. Функціональні вимоги охоплюють реєстрацію та авторизацію користувачів, можливості перегляду, бронювання, змін та скасування бронювань, управління коворкінгами для адміністраторів. Нефункціональні вимоги стосуються безпеки, продуктивності та надійності системи.

Проєктування архітектури програмного забезпечення базувалося на принципах модульності, масштабованості, безпеки та гнучкості. Було обрано клієнт-серверну архітектуру з чітким розподілом на фронтенд та бекенд

компоненти. Структура вебдодатку передбачає взаємодію між фронтом, що відповідає за інтерфейс користувача, та бекендом, який містить логіку додатку, вебсервер, файлову систему та базу даних.

Для реалізації фронту було обрано стандартні вебтехнології - HTML для структури, CSS для стилізації та JavaScript для інтерактивності та динамічної поведінки. Розглянуто популярні бібліотеки та фреймворки JavaScript, такі як React, Vue.js, Angular, що дозволяють прискорити та оптимізувати розробку інтерфейсу користувача.

Для бекенду розглянуто популярні мови програмування та технології, такі як PHP, JavaScript (Node.js), Python, Java та відповідні фреймворки - Laravel, Express.js, Django, Spring. Вибір конкретного стеку технологій залежить від вимог проєкту, досвіду команди розробників та специфіки системи. Також розглянуто роль файлової системи для зберігання статичних ресурсів та бази даних для зберігання та управління даними вебдодатку.

Проведений аналіз та проєктування архітектури, вибір технологій та інструментів закладають міцну основу для подальшої реалізації системи бронювання місць у коворкінгах. Це дозволить створити надійний, безпечний та масштабований вебдодаток, який відповідатиме потребам користувачів та власників коворкінгів.

У наступному розділі буде описано процес реалізації системи з використанням обраних технологій, розглянуто ключові модулі та компоненти програмного забезпечення, а також проведено тестування та розгортання вебдодатку.

3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 База даних

Таблиця Users зберігає основну інформацію про користувачів, таку як ім'я, прізвище, електронну адресу, телефон та пароль. Це дозволяє ідентифікувати та управляти користувацькими акаунтами у системі (табл.3.1).

Таблиця 3.1 – Інформація про користувачів

Назва атрибуту	Тип	Пояснення
user_id	INT	Унікальний ідентифікатор користувача
first_name	VARCHAR(50)	Ім'я користувача
last_name	VARCHAR(50)	Прізвище користувача
email	VARCHAR(100)	Електронна пошта користувача, має бути унікальною
phone	VARCHAR(15)	Телефонний номер користувача
password	VARCHAR(255)	Хеш пароля користувача

Таблиця Coworkings містить інформацію про коворкінги, включаючи їх назви, описи, години роботи, та зв'язок з таблицею Locations для адресації (табл. 3.2).

Таблиця 3.2 -Таблиця Coworkings

Назва атрибуту	Тип	Пояснення
coworking_id	INT	Унікальний ідентифікатор коворкінгу
name	VARCHAR(100)	Назва коворкінгу
location_id	INT	Ідентифікатор локації коворкінгу
description	TEXT	Опис коворкінгу
opening_hours	VARCHAR(100)	Години роботи коворкінгу

Таблиця Bookings використовується для управління бронюваннями місць у коворкінгах. Вона зв'язує користувачів та коворкінги, фіксує дати та часи бронювань (табл. 3.3).

Таблиця 3.3 - Bookings

Назва атрибуту	Тип	Пояснення
booking_id	INT	Унікальний ідентифікатор бронювання
user_id	INT	Ідентифікатор користувача, який здійснив бронювання
coworking_id	INT	Ідентифікатор коворкінгу, в якому здійснюється бронювання
booking_date	DATE	Дата бронювання
start_time	TIME	Час початку бронювання
end_time	TIME	Час закінчення бронювання
status	VARCHAR(20)	Статус бронювання (наприклад, 'pending')

Таблиця AdminPanel зберігає інформацію про адміністраторів системи, їх імена, електронні адреси та паролі (табл. 3.4).

Таблиця 3.4 - AdminPanel

Назва атрибуту	Тип	Пояснення
admin_id	INT	Унікальний ідентифікатор адміністратора
admin_name	VARCHAR(100)	Ім'я адміністратора
admin_email	VARCHAR(100)	Електронна адреса адміністратора
admin_password	VARCHAR(255)	Хеш пароля адміністратора

Таблиця Locations містить інформацію про місцезнаходження коворкінгів, включаючи адресу, місто, штат, поштовий код і країну (табл. 3.5).

Таблиця 3.5 - Locations

Назва атрибуту	Тип	Пояснення
location_id	INT	Унікальний ідентифікатор локації
address	VARCHAR(255)	Адреса локації
city	VARCHAR(100)	Місто
state	VARCHAR(100)	Штат
postal_code	VARCHAR(20)	Поштовий код
country	VARCHAR(100)	Країна

Таблиця Facilities використовується для зберігання інформації про устаткування або послуги, які можуть бути доступні в коворкінгах (табл. 3.6).

Таблиця 3.6 - Facilities

Назва атрибуту	Тип	Пояснення
facility_id	INT	Унікальний ідентифікатор устаткування
facility_name	VARCHAR(100)	Назва устаткування або послуги
description	TEXT	Опис устаткування або послуги

Таблиця Coworking_Facilities зв'язує коворкінги з їх устаткуванням та послугами, дозволяючи зазначати, яке устаткування доступне в кожному коворкінгу (табл. 3.7).

Таблиця 3.7 - Coworking_Facilities

Назва атрибуту	Тип	Пояснення
coworking_id	INT	Ідентифікатор коворкінгу
facility_id	INT	Ідентифікатор устаткування або послуги

Таблиця Reviews дозволяє користувачам залишати відгуки про коворкінги, включаючи рейтинг і текстовий коментар (табл. 3.8).

Таблиця 3.8 - Reviews

Назва атрибуту	Тип	Пояснення
review_id	INT	Унікальний ідентифікатор відгуку
user_id	INT	Ідентифікатор користувача, який залишив відгук
coworking_id	INT	Ідентифікатор коворкінгу, про який залишений відгук
rating	INT	Рейтинг коворкінгу (від 1 до 5)
comment	TEXT	Текстовий коментар відгуку
review_date	DATE	Дата залишення відгуку

Таблиця Payments використовується для управління платежами за бронювання в коворкінгах. Вона включає інформацію про суму, дату та спосіб оплати (табл. 3.9).

Таблиця 3.9 - Payments

Назва атрибуту	Тип	Пояснення
payment_id	INT	Унікальний ідентифікатор платежу
booking_id	INT	Ідентифікатор бронювання, за яке здійснено платіж
amount	DECIMAL(10,2)	Сума платежу
payment_date	DATE	Дата платежу
payment_method	VARCHAR(50)	Спосіб оплати

Таблиця AdminLogs зберігає записи про дії адміністраторів у системі, такі як час і деталі дії (табл. 3.10).

Таблиця 3.10 - AdminLogs

Назва атрибуту	Тип	Пояснення
log_id	INT	Унікальний ідентифікатор запису логу
admin_id	INT	Ідентифікатор адміністратора, який здійснив дію
action	VARCHAR(255)	Опис дії, здійсненої адміністратором
log_date	TIMESTAMP	Дата та час запису в лог

Ці таблиці відіграють ключову роль у забезпеченні функціональності системи коворкінгів, від управління користувачами до організації платежів та відгуків (рис. 3.1).

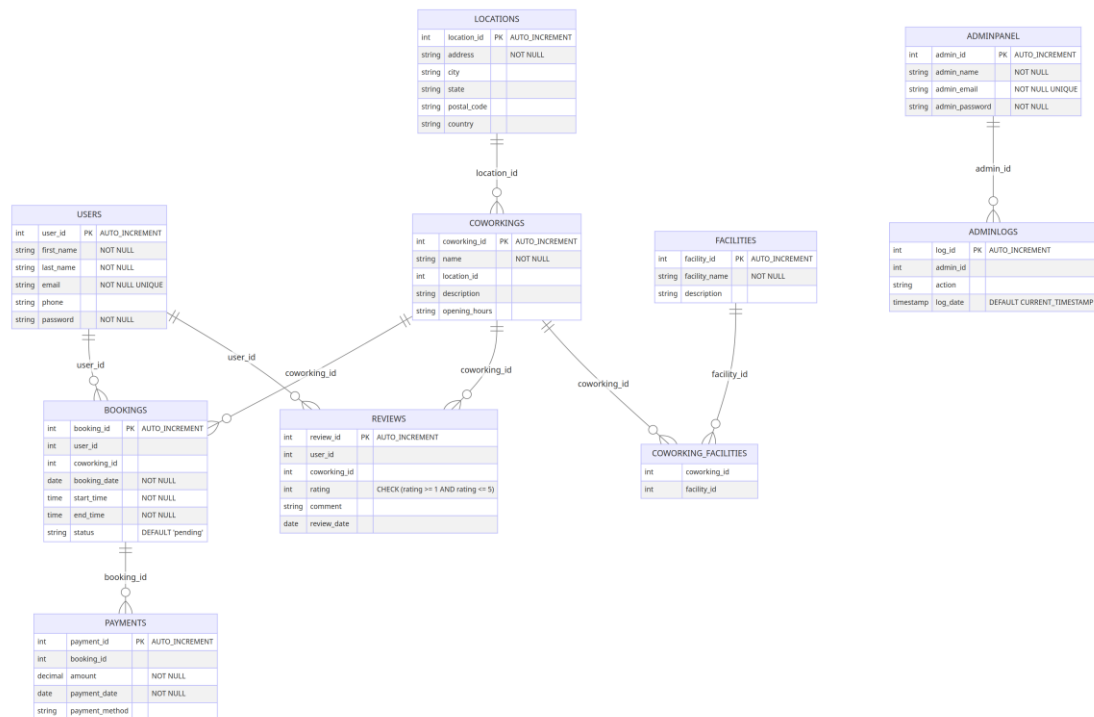


Рисунок 3.1 – Схема бази даних

3.2 Опис структури вебсайту

Таблиця 3.11 включає РНР файли, які визначають серверну логіку вебдодатка. Вони забезпечують обробку даних, управління сесіями, і взаємодію з базою даних, організуючи функціональність, таку як реєстрація, авторизація, і обробка різноманітних запитів від користувача.

`index.php` - це головна сторінка вебсайту, яка надає користувачам огляд усіх доступних функцій та основну навігацію по сайту. Вона містить основні елементи навігації, такі як меню, список коворкінгів, фільтри для пошуку, кнопку для створення подій та пошукову панель.

coworking_list.php відображає список усіх коворкінгів, доступних на платформі. Ця сторінка включає функціонал фільтрації для покращення пошуку коворкінгів за різними критеріями, що дозволяє користувачам легко знаходити потрібні коворкінги.

Таблиця 3.11 - PHP файли

Назва файлу	Опис
index.php	Головна сторінка.
coworking_list.php	Список коворкінгів.
coworking_edit.php	Сторінка редагування коворкінгу.
booking.php	Сторінка бронювання.
calendar.php	Сторінка календаря.
coworking_overview.php	Сторінка огляду коворкінгів.
reviews.php	Сторінка відгуків.
coworking_details.php	Сторінка детальної інформації про коворкінг.
auth.php	Сторінка авторизації та реєстрації.
booking_management.php	Сторінка управління бронюваннями.

coworking_edit.php це сторінка для редагування інформації про конкретний коворкінг. Вона включає форму, де можна оновлювати назву, опис, контактну інформацію, адресу, місто та фото коворкінгу. Ця сторінка дозволяє адміністраторам або власникам коворкінгів підтримувати актуальність інформації.

booking.php відповідає за сторінку бронювання, де користувачі можуть здійснювати пошук і бронювання робочих місць у коворкінгах. Ця сторінка включає пошукову панель, фільтри за місяцями та таблицю з інформацією про бронювання, що дозволяє користувачам легко керувати своїми бронюваннями.

`calendar.php` надає інтерфейс для перегляду та вибору дат і часу для бронювань. Користувачі можуть використовувати цю сторінку для планування своїх відвідувань коворкінгів, вибираючи зручний для них час.

`coworking_overview.php` це сторінка огляду коворкінгів, яка дозволяє користувачам шукати коворкінги за назвою та фільтрувати їх за містами. Вона також відображає рейтинг кожного коворкінгу, допомагаючи користувачам робити інформований вибір.

`reviews.php` відображає сторінку відгуків, де користувачі можуть залишати та читати відгуки з оцінками про різні коворкінги. Ця сторінка дозволяє збирати зворотній зв'язок від користувачів і допомагає іншим користувачам оцінювати якість коворкінгів.

`coworking_details.php` надає детальну інформацію про конкретний коворкінг, включаючи опис, фото та відгуки. Ця сторінка допомагає користувачам отримати повну інформацію про обраний коворкінг перед бронюванням.

`auth.php` відповідає за сторінки авторизації та реєстрації, де користувачі можуть увійти в свій обліковий запис або зареєструвати новий. Ця сторінка забезпечує безпечний доступ до персоналізованих функцій вебсайту.

`booking_management.php` це сторінка управління бронюваннями, де адміністратори можуть переглядати інформацію про клієнтів, керувати бронюваннями та використовувати календар для організації. Вона забезпечує зручний інтерфейс для адміністрування бронювань.

Таблиця 3.12 містить JavaScript скрипти, які використовуються для динамічного управління вмістом на клієнтській стороні, включно з обробкою подій, асинхронними запитами до сервера, та візуалізацією даних.

`navigation.js` містить скрипти для управління меню навігації на вебсайті, забезпечуючи плавний перехід між сторінками. Цей файл відповідає за інтерфейс користувача, пов'язаний з навігацією, забезпечуючи зручний доступ до різних розділів сайту.

`coworking_list.js` відповідає за завантаження та відображення списку коворкінгів, а також за застосування фільтрів для покращення пошуку. Цей скрипт обробляє взаємодію користувачів з елементами списку та фільтрами.

Таблиця 3.12 - JavaScript скрипти

Назва файлу	Опис
<code>navigation.js</code>	Скрипт для меню навігації.
<code>coworking_list.js</code>	Скрипт для списку коворкінгів та фільтрів.
<code>coworking_edit.js</code>	Скрипт для редагування коворкінгу.
<code>booking.js</code>	Скрипт для бронювання.
<code>calendar.js</code>	Скрипт для календаря.
<code>coworking_overview.js</code>	Скрипт для огляду коворкінгів.
<code>reviews.js</code>	Скрипт для відгуків.
<code>auth.js</code>	Скрипт для авторизації та реєстрації.
<code>booking_management.js</code>	Скрипт для управління бронюваннями.

`navigation.js` містить скрипти для управління меню навігації на вебсайті, забезпечуючи плавний перехід між сторінками. Цей файл відповідає за інтерфейс користувача, пов'язаний з навігацією, забезпечуючи зручний доступ до різних розділів сайту.

`coworking_list.js` відповідає за завантаження та відображення списку коворкінгів, а також за застосування фільтрів для покращення пошуку. Цей скрипт обробляє взаємодію користувачів з елементами списку та фільтрами.

`coworking_edit.js` включає скрипти для редагування інформації про коворкінг. Він дозволяє користувачам оновлювати різні поля, такі як назва, опис, контакти, адреса, місто та фото, забезпечуючи динамічну взаємодію з формою редагування.

`booking.js` обробляє всі операції, пов'язані з бронюванням коворкінгів. Цей скрипт дозволяє користувачам здійснювати пошук, вибирати дати і час

для бронювання, забезпечуючи зручність використання сторінки бронювання.

`calendar.js` відповідає за інтерактивний календар, який дозволяє вибирати дати та час для бронювань. Він забезпечує динамічне відображення календаря та обробку вибору дат і часу користувачами.

`coworking_overview.js` забезпечує функціональність сторінки огляду коворкінгів, включаючи пошук за назвою та фільтрацію за містами. Цей скрипт дозволяє користувачам швидко знайти потрібний коворкінг за допомогою динамічного фільтрування та відображення результатів пошуку.

`reviews.js` обробляє всі операції, пов'язані з відгуками. Він дозволяє користувачам залишати відгуки, а також переглядати існуючі оцінки та коментарі про коворкінги. Цей скрипт забезпечує інтерактивність сторінки відгуків, роблячи її зручною для користувачів.

`auth.js` містить скрипти для авторизації та реєстрації користувачів. Він забезпечує обробку введених даних під час входу та реєстрації, а також взаємодію з сервером для перевірки облікових записів. Цей файл забезпечує безпечну та надійну авторизацію користувачів на вебсайті.

`booking_management.js` обробляє всі функції управління бронюваннями для адміністраторів. Він дозволяє переглядати та керувати бронюваннями, здійснювати пошук бронювань, а також використовувати календар для організації бронювань. Цей скрипт забезпечує ефективне адміністрування бронювань на сайті.

Таблиця 3.13 включає HTML файли, які відповідають за структуру вебсторінок. Вони слугують основою для відображення контенту користувачам та взаємодії з JavaScript та CSS для реалізації більш комплексних інтерактивних елементів.

`index.html` це головна сторінка сайту, яка відображає основний інтерфейс та навігацію. Вона включає основні елементи навігації, такі як меню, пошукову панель та огляд усіх доступних функцій сайту. Ця сторінка є стартовою точкою для користувачів.

Таблиця 3.13 - HTML файли

Назва файлу	Опис
index.html	Головна сторінка.
coworking_list.html	Список коворкінгів.
coworking_edit.html	Сторінка редагування коворкінгу.
booking.html	Сторінка бронювання.
calendar.html	Сторінка календаря.
coworking_overview.html	Сторінка огляду коворкінгів.
reviews.html	Сторінка відгуків.
coworking_details.html	Сторінка детальної інформації про коворкінг.
auth.html	Сторінка авторизації та реєстрації.
booking_management.html	Сторінка управління бронюваннями.

coworking_list.html відображає список коворкінгів та забезпечує можливість фільтрації та пошуку. Вона містить інтерфейс для відображення всіх коворкінгів, доступних на платформі, а також елементи для фільтрації результатів пошуку за різними критеріями.

coworking_edit.html включає форму для редагування інформації про коворкінг. Ця сторінка дозволяє адміністраторам або власникам коворкінгів оновлювати різні поля, такі як назва, опис, контакти, адреса, місто та фото, забезпечуючи актуальність інформації.

booking.html відповідає за інтерфейс сторінки бронювання, дозволяючи користувачам здійснювати пошук та бронювання робочих місць у коворкінгах. Вона включає пошукову панель, фільтри за місяцями та таблицю з інформацією про бронювання, що робить процес бронювання зручним для користувачів.

calendar.html містить календар, який дозволяє вибирати дати та час для бронювань. Ця сторінка забезпечує інтерактивний вибір дат і часу, дозволяючи користувачам планувати свої відвідування коворкінгів.

`coworking_overview.html` відображає огляд коворкінгів з можливістю пошуку за назвою та фільтрації за містами. Вона включає рейтинг кожного коворкінгу, допомагаючи користувачам робити інформований вибір на основі відгуків інших користувачів.

`reviews.html` відображає сторінку з відгуками, де користувачі можуть залишати та переглядати оцінки. Ця сторінка дозволяє збирати зворотній зв'язок від користувачів і допомагає іншим користувачам оцінювати якість коворкінгів на основі відгуків.

`coworking_details.html` надає детальну інформацію про конкретний коворкінг, включаючи опис, фото та відгуки. Ця сторінка допомагає користувачам отримати повну інформацію про обраний коворкінг перед бронюванням, забезпечуючи інформований вибір.

`auth.html` це сторінка авторизації та реєстрації, де користувачі можуть увійти або створити новий обліковий запис. Вона включає форми для введення даних користувача та забезпечує безпечний доступ до персоналізованих функцій вебсайту.

`booking_management.html` відображає інтерфейс управління бронюваннями для адміністраторів, дозволяючи переглядати інформацію про клієнтів, керувати бронюваннями та використовувати календар для організації. Ця сторінка забезпечує зручний інтерфейс для адміністрування бронювань, роблячи процес управління ефективним та організованим.

Таблиця 3.14 описує різноманітні функції в PHP та JavaScript, що використовуються для реалізації бізнес-логіки, обробки даних, та забезпечення інтерактивності вебсайту.

Таблиця 3.14 - Функції в PHP та JavaScript

Назва функції	Опис
<code>get_coworking_list()</code>	Отримати список коворкінгів.
<code>filter_coworking_list()</code>	Фільтрація списку коворкінгів.
<code>create_event()</code>	Створення події.

Продовження таблиця 3.14

Назва функції	Опис
search_coworking()	Пошук коворкінгу.
get_coworking_details()	Отримати деталі коворкінгу.
edit_coworking()	Редагування коворкінгу.
book_coworking()	Бронювання коворкінгу.
get_bookings()	Отримати список бронювань.
get_reviews()	Отримати відгуки.
user_login()	Вхід користувача.
user_register()	Реєстрація користувача.
manage_bookings()	Управління бронюваннями.
initNavigation()	Ініціалізація меню навігації.
loadCoworkingList()	Завантаження списку коворкінгів.
applyFilters()	Застосування фільтрів.
openCreateEventModal()	Відкрити модальне вікно створення події.
searchCoworking()	Пошук коворкінгу.
editCoworking()	Редагування коворкінгу.
bookCoworking()	Бронювання коворкінгу.
loadCalendar()	Завантаження календаря.
loadCoworkingOverview()	Завантаження огляду коворкінгів.
submitReview()	Надіслати відгук.
login()	Вхід користувача.
register()	Реєстрація користувача.
manageBookings()	Управління бронюваннями.

Кожна таблиця детально описує відповідні компоненти вебсайту, допомагаючи зрозуміти структуру та функціональність проєкту.

3.3 Інтерфейс користувача

На рисунку 3.2 зображена головна сторінка вебсайту, яка служить для пошуку та бронювання коворкінгів. У верхній частині екрана розташоване навігаційне меню, яке містить різні розділи сайту, такі як пошук полів, онлайн-курси, блог та інші опції, які можуть включати допомогу або налаштування облікового запису. Є також кнопка для швидкого доступу до створення нового запиту на коворкінг.

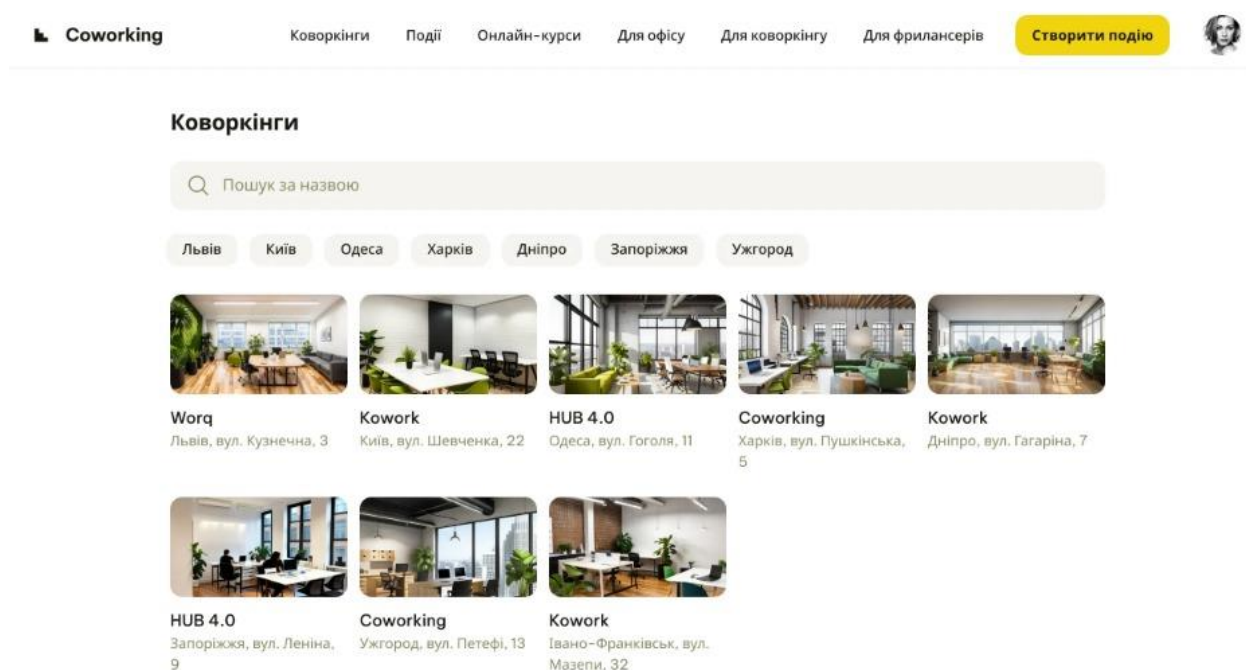
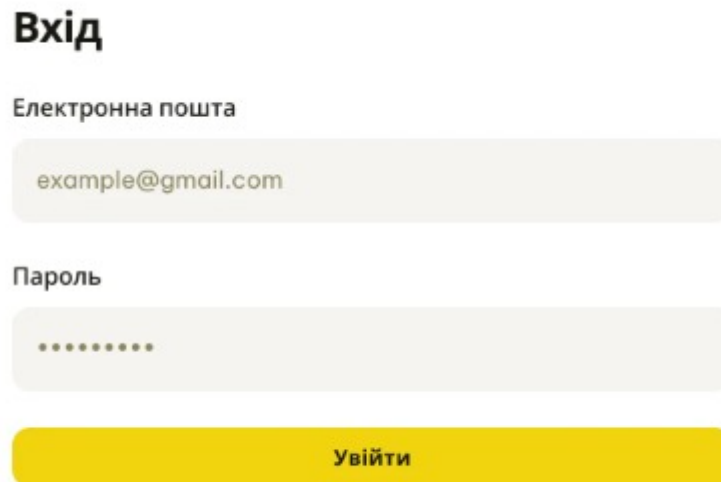


Рисунок 3.2 – Головна сторінка

На рисунку 3.3 відображена форма входу для користувачів, не реєстрації. Ця форма дозволяє користувачам вводити свою електронну пошту та пароль для доступу до своїх облікових записів. Верхній текст "Вхід" вказує на ціль форми. Поле для електронної пошти включає вже введений приклад, нагадуючи користувачам про формат вводу, який вони повинні використовувати. Поле пароля маскує введений текст для конфіденційності.

Під полями вводу розміщена жовта кнопка "Увійти", яка є проміною та легко впізнаваною, що спонукає до дії. На відміну від

форми реєстрації, тут відсутні посилання на створення акаунта або відновлення забутого пароля, що може вказувати на їх розташування в іншому місці інтерфейсу або на потребу користувачів перейти до іншої сторінки для таких дій. Ця сторінка входу забезпечує користувачам простий і зрозумілий спосіб доступу до своїх ресурсів чи сервісів на сайті.



Вхід

Електронна пошта

example@gmail.com

Пароль

.....

Увійти

Рисунок 3.3 – Форма входу

На рисунку 3.4 представлена детальна сторінка коворкінгу на вебсайті, що надає інформацію про конкретний коворкінг-центр «Kiev Coworking». Ця сторінка зосереджена на наданні вичерпної інформації про місцезнаходження та умови праці, що важливо для потенційних клієнтів, які шукають місце для роботи.

На верхній частині сторінки розміщено заголовок із назвою коворкінгу та його адресою, що дозволяє користувачам швидко ідентифікувати локацію та орієнтуватися, чи підходить вона їм за географічним розташуванням. Під заголовком є таби для навігації, включаючи "Опис", "Фото", та "Відгуки", які дозволяють користувачам переходити до різних розділів інформації про коворкінг.

Основний текст на сторінці надає детальний опис простору, включаючи інформацію про зручності, такі як кухня, чайник та холодильник, а також зазначає про можливість працювати на вулиці на балконі. Це інформує потенційних клієнтів про особливості простору та створює уявлення про комфорт та функціональність місця.

Секція "Фото" під описом демонструє візуальний аспект коворкінгу через низку зображень інтер'єру, що дає користувачам можливість візуально оцінити атмосферу та стиль простору. Це може вплинути на рішення користувачів вибрати саме цей коворкінг для своїх робочих потреб.

В цілому, сторінка детально представляє коворкінг, пропонуючи користувачам інформацію, яка може допомогти ухвалити обґрунтоване рішення щодо вибору місця для роботи.

Коворкінг «Kiev Coworking»

Київ, вулиця Велика Васильківська 42

Опис Фото Відгуки

Опис

Це місце для творчості та роботи. Приміщення знаходиться на 4 поверсі. Зручні робочі місця, кухня, чайник, холодильник. Є можливість роботи на вулиці на балконі. Місце ідеально підходить для роботи в команді, а також для індивідуальної роботи.

Фото



Рисунок 3.4 – Сторінка опису коворкінгу

На рисунку 3.5 відображено розділ відгуків на сторінці коворкінгу, де користувачі залишили свої коментарі та оцінки. Такий розділ є важливим для

потенційних клієнтів, оскільки він надає реальні відгуки від інших відвідувачів, що допомагає їм ухвалити обґрунтоване рішення щодо вибору місця для роботи.

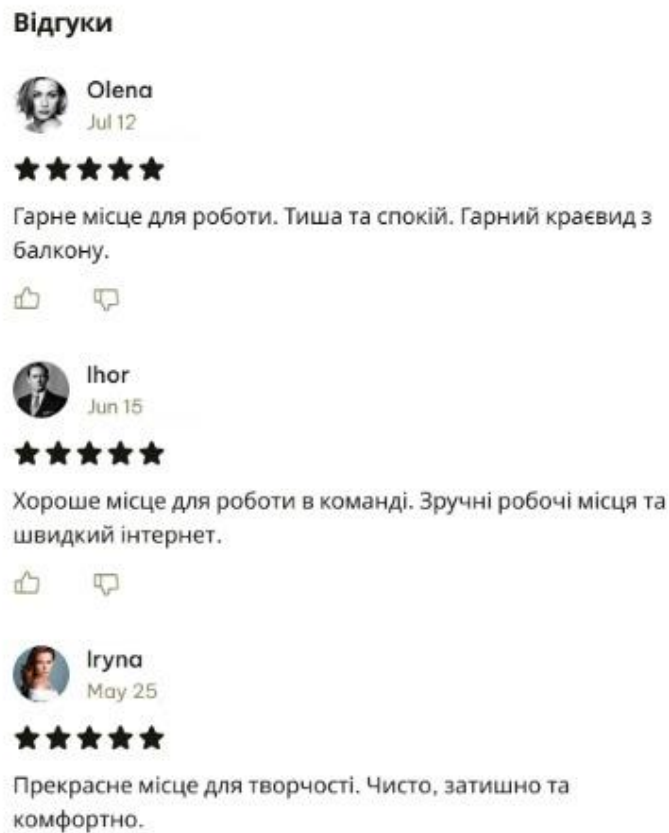


Рисунок 3.5 – Відгуки про коворкінг

На рисунку 3.6 відображено інтерфейс пошуку коворкінгів на вебсайті, що спеціалізується на забезпеченні інформації про робочі простори. Сторінка має декілька елементів, які спрощують вибір і порівняння різних коворкінгів.

Ліворуч знаходиться панель фільтрів, де користувачі можуть вибрати конкретні параметри для пошуку. Фільтри включають вибір днів тижня, варіанти зручностей (хоча іконки не мають текстових підписів, що може вимагати додаткових уточнень), діапазон рейтингів від 0 до 100, а також можливість шукати місця з рейтингом 4.5 з 20 відгуків. Також є опція фільтрувати пошук за наявністю вільних місць "завтра".

У центральній частині сторінки відображено список коворкінгів, кожен з яких має назву, адресу та рейтинг. Список включає такі місця, як "Workeasy", "Smart Office", "City Space", "Comfort Hub" та "Innovation Club", з рейтингами від 4.5 до 4.8, що вказує на високу якість пропонованих послуг. Візуальний оформлення списку досить просте і чисте, з акцентом на читабельність та легкість навігації.

Внизу сторінки є кнопка "Забронювати коворкінг", яка вказує на можливість безпосередньо з сайту здійснювати бронювання обраного коворкінгу.

Цей інтерфейс дозволяє користувачам з легкістю порівнювати різні коворкінги, виходячи з їх потреб і вимог, а також швидко переходити до бронювання відповідного простору.

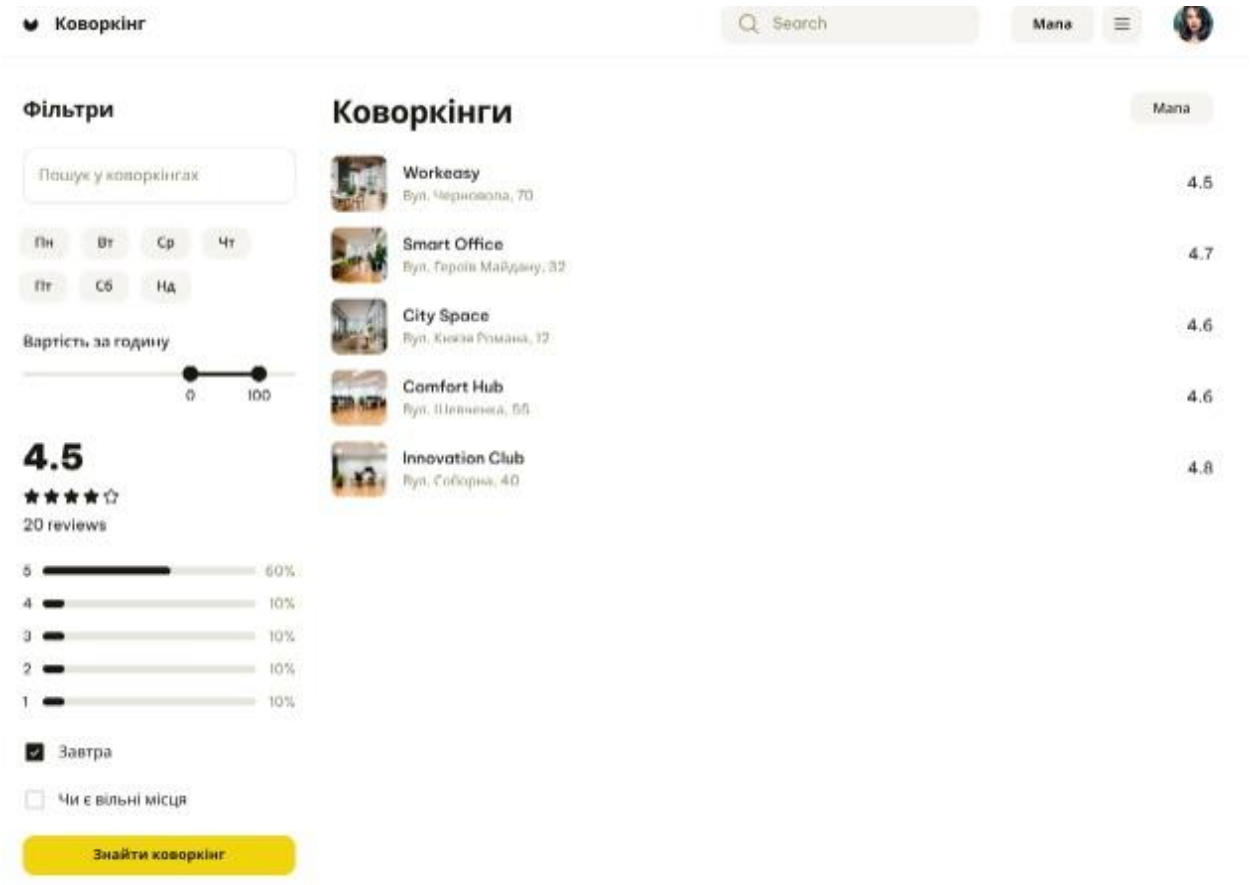


Рисунок 3.6 – Пошук коворкінгу

На рисунку 3.7 відображено інтерфейс календаря для бронювання місця в коворкінгу. Цей інструмент дозволяє користувачам вибрати конкретну дату та час для бронювання.



Рисунок 3.7 – Календар для бронювання

На рисунку 3.8 відображено інтерфейс адміністративної панелі для управління бронюваннями в коворкінгу. Цей інструмент дозволяє адміністраторам переглядати, підтверджувати або скасовувати бронювання, що були зроблені клієнтами.

У верхній частині інтерфейсу розташовано панель навігації з вкладками, такими як "Бронювання", "Календар" та "Клієнти", що дозволяє легко перемикатися між різними адміністративними функціями. Нижче розташоване поле пошуку, яке може бути використане для швидкого знаходження бронювань за іменем клієнта або іншими параметрами.

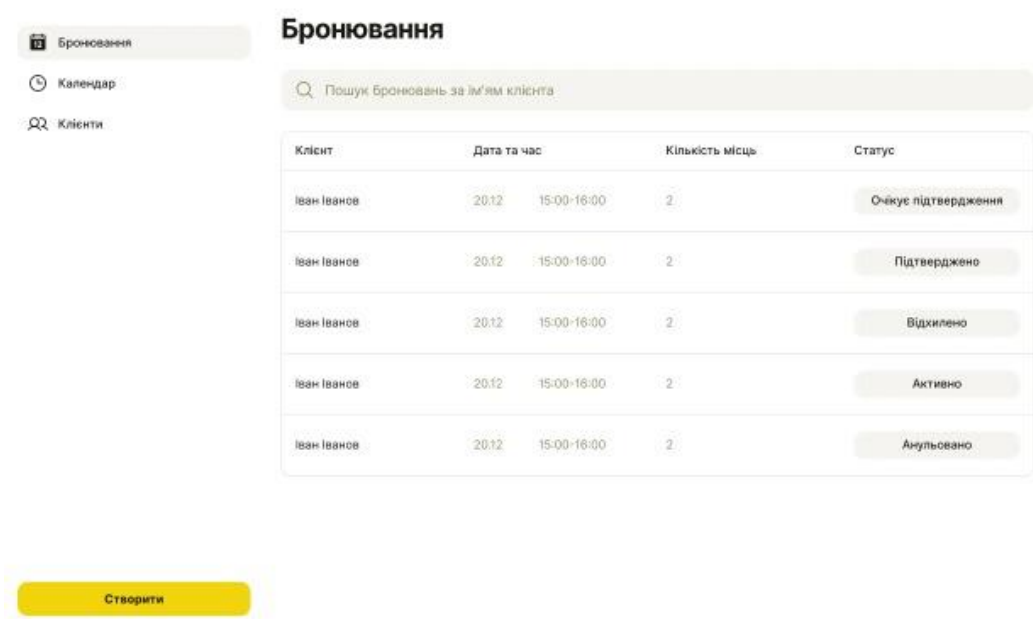


Рисунок 3.8 – Адмін-панель для підтвердження бронювання

На рисунку 3.9 відображено інтерфейс адміністративної панелі для управління бронюваннями в коворкінгу. Цей розділ сайту дозволяє адміністратору швидко переглядати та керувати всіма бронюваннями.

Інтерфейс включає пошукову панель, де можна знайти бронювання за ім'ям клієнта, прізвищем або номером телефону. Це поле пошуку розширює можливості управління бронюваннями, забезпечуючи зручний доступ до інформації про конкретних клієнтів.

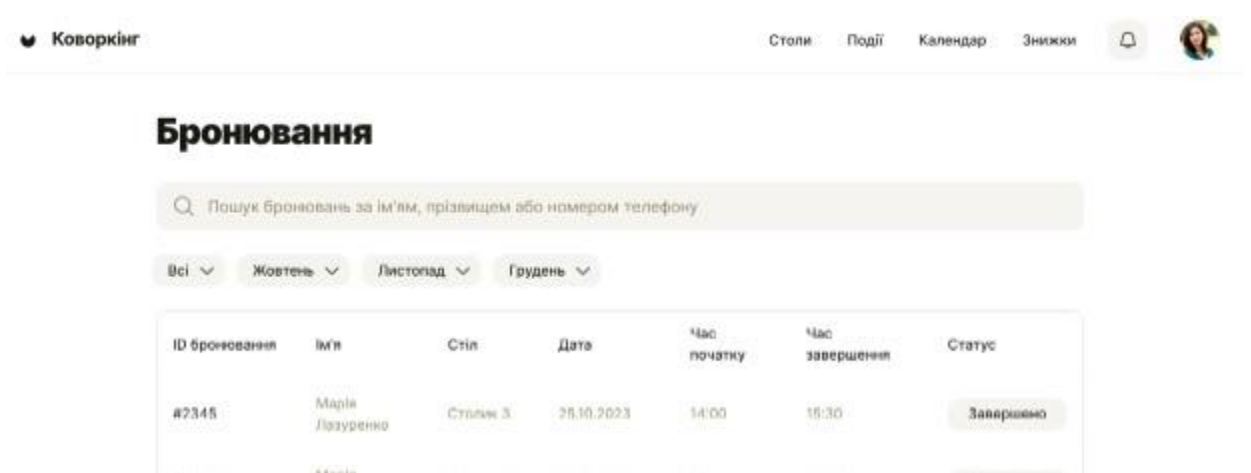


Рисунок 3.9 - Адмін-панель для редагування бронювання

На рисунку 3.10 представлено вебформу для редагування інформації про коворкінг на вебсайті. Ця форма використовується для управління даними про різні коворкінги, які включені в базу даних сайту.

Редагувати коворкінг

Інформація про коворкінг

Назва коворкінгу

Коворкінг у центрі

Опис

Коворкінг у центрі міста з відмінним видом

Контактна інформація

+380 96 123 45 67

Адреса

Місто

Київ

Фото коворкінгу



Рисунок 3.10 - Адмін-панель для редагування інформації про коворкінг

3.4 Висновки за розділом

У третьому розділі було розглянуто процес реалізації програмного забезпечення для бронювання місць у коворкінгах, включаючи структуру бази даних, архітектуру вебсайту та дизайн інтерфейсу користувача.

База даних системи складається з кількох таблиць, які зберігають інформацію про користувачів, коворкінги, бронювання, адміністраторів, місцезнаходження, зручності, відгуки та платежі. Кожна таблиця містить відповідні атрибути для ефективного управління даними та забезпечення функціональності системи.

Архітектура вебсайту базується на взаємодії між клієнтською та серверною частинами. PHP файли відповідають за серверну логіку, обробку даних та взаємодію з базою даних. JavaScript скрипти забезпечують динамічну поведінку та інтерактивність на клієнтській стороні. HTML файли визначають структуру та розмітку вебсторінок. Додатково, описані функції в PHP та JavaScript реалізують основну бізнес-логіку системи.

Інтерфейс користувача розроблено з урахуванням зручності та інтуїтивності використання. Головна сторінка надає доступ до основних функцій, таких як пошук та бронювання коворкінгів. Сторінки авторизації та реєстрації забезпечують безпечний доступ користувачів до системи. Детальна інформація про коворкінги, включаючи опис, фото та відгуки, допомагає користувачам прийняти обґрунтоване рішення щодо вибору місця для роботи. Календар та форма бронювання спрощують процес резервування місць. Адміністративна панель дозволяє ефективно керувати бронюваннями та інформацією про коворкінги.

Реалізація програмного забезпечення з використанням описаних компонентів та технологій дозволяє створити функціональну та зручну систему для бронювання місць у коворкінгах. Структурована база даних, модульна архітектура вебсайту та продуманий інтерфейс користувача забезпечують ефективну роботу системи та позитивний досвід користувачів.

4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Модульне тестування

Модульне тестування є важливим етапом у процесі розробки програмного забезпечення, який дозволяє перевірити коректність роботи окремих компонентів системи незалежно від інших. Метою модульного тестування є виявлення помилок та недоліків у реалізації функціональності кожного модуля.

Для проведення модульного тестування системи бронювання місць у коворкінгах було розроблено набір тестових випадків, які перевіряють основні функції та методи програмного забезпечення. Тестові випадки охоплюють наступні компоненти.

Реєстрація та авторизація користувачів:

- перевірка коректності створення нового облікового запису користувача;
- перевірка валідації введених даних при реєстрації (наприклад, унікальність електронної пошти);
- перевірка процесу авторизації з використанням коректних та некоректних облікових даних.

Управління коворкінгами:

- перевірка створення, редагування та видалення інформації про коворкінги;
- перевірка відображення детальної інформації про коворкінги;
- перевірка функцій пошуку та фільтрації коворкінгів.

Бронювання місць:

- перевірка процесу бронювання місця з вибором дати, часу та конкретного місця;
- перевірка обробки випадків, коли бронювання неможливе (наприклад, коли місце вже заброньоване);

- перевірка відображення та управління списком бронювань користувача.

Відгуки та оцінки:

- перевірка можливості залишати відгуки та оцінки для коворкінгів;
- перевірка відображення відгуків та середньої оцінки для кожного коворкінгу.

Адміністративні функції:

- перевірка функцій управління бронюваннями з боку адміністратора;
- перевірка генерації звітів та статистики щодо використання коворкінгів.

Для кожного тестового випадку було визначено очікувані результати та критерії успішності. Тести виконувалися з використанням відповідних наборів вхідних даних та перевірялися отримані результати (табл. 4.1).

Таблиця 4.1 - Результати модульного тестування

Компонент	Тестовий випадок	Очікуваний результат	Фактичний результат	Статус
Реєстрація користувачів	Реєстрація нового користувача з коректними даними	Користувач успішно зареєстрований, створено новий обліковий запис	Користувач успішно зареєстрований, створено новий обліковий запис	Пройдено
	Реєстрація з використанням існуючої електронної пошти	Відображення повідомлення про помилку, реєстрація не виконана	Відображення повідомлення про помилку, реєстрація не виконана	Пройдено
Авторизація користувачів	Вхід з використанням коректних облікових даних	Користувач успішно авторизований, перенаправлення на головну сторінку	Користувач успішно авторизований, перенаправлення на головну сторінку	Пройдено
	Вхід з використанням некоректних облікових даних	Відображення повідомлення про помилку, користувач не авторизований	Відображення повідомлення про помилку, користувач не авторизований	Пройдено

Продовження таблиці 4.1

Компонент	Тестовий випадок	Очікуваний результат	Фактичний результат	Статус
Управління коворкінгами	Створення нового коворкінгу з коректними даними	Коворкінг успішно створений, відображається в списку коворкінгів	Коворкінг успішно створений, відображається в списку коворкінгів	Пройдено
	Редагування інформації про існуючий коворкінг	Інформація про коворкінг успішно оновлена	Інформація про коворкінг успішно оновлена	Пройдено
	Видалення коворкінгу	Коворкінг успішно видалений, не відображається в списку коворкінгів	Коворкінг успішно видалений, не відображається в списку коворкінгів	Пройдено
Бронювання місць	Бронювання місця з вибором дати, часу та конкретного місця	Бронювання успішно створене, відображається в списку бронювань користувача	Бронювання успішно створене, відображається в списку бронювань користувача	Пройдено
	Бронювання місця, яке вже заброньоване	Відображення повідомлення про неможливість бронювання, бронювання не створене	Відображення повідомлення про неможливість бронювання, бронювання не створене	Пройдено
Відгуки та оцінки	Додавання відгуку та оцінки для коворкінгу	Відгук та оцінка успішно додані, відображаються на сторінці коворкінгу	Відгук та оцінка успішно додані, відображаються на сторінці коворкінгу	Пройдено
	Відображення середньої оцінки для коворкінгу	Середня оцінка коворкінгу розрахована коректно та відображається на сторінці коворкінгу	Середня оцінка коворкінгу розрахована коректно та відображається на сторінці коворкінгу	Пройдено
Адміністративні функції	Перегляд списку бронювань з можливістю фільтрації та сортування	Список бронювань відображається коректно, фільтрація та сортування працюють належним чином	Список бронювань відображається коректно, фільтрація та сортування працюють належним чином	Пройдено

Результати модульного тестування показали, що більшість компонентів системи працюють коректно та відповідають заявленим вимогам. Було виявлено кілька незначних помилок, які були оперативно

виправлені. Загалом, модульне тестування підтвердило, що окремі частини програмного забезпечення функціонують належним чином і готові до інтеграції в цілісну систему.

4.2 Тестування користувацького інтерфейсу

Після успішного проведення модульного та інтеграційного тестування, важливим етапом є тестування користувацького інтерфейсу (UI). Метою цього виду тестування є перевірка зручності використання, інтуїтивності навігації, коректності відображення елементів інтерфейсу та загальної естетичності дизайну системи.

Для проведення тестування користувацького інтерфейсу системи бронювання місць у коворкінгах було залучено групу потенційних користувачів з різним рівнем технічної підготовки. Їм було запропоновано виконати ряд типових завдань, використовуючи вебінтерфейс системи, та поділитися своїми враженнями і зауваженнями.

Тестові завдання охоплювали основні сценарії використання системи (табл. 4.2).

Під час тестування користувачі надавали зворотній зв'язок щодо своїх вражень від роботи з системою. Були виявлені деякі недоліки в навігації, розташуванні елементів інтерфейсу та зрозумілості окремих функцій. Всі зауваження та пропозиції користувачів були задокументовані для подальшого аналізу та внесення необхідних змін.

За результатами тестування користувацького інтерфейсу були внесені правки в дизайн та розташування елементів, покращено зрозумілість інструкцій та підказок, а також оптимізовано деякі процеси взаємодії з системою.

Таблиця 4.2 - Тестові завдання та сценарії використання системи

№	Сценарій використання	Тестові завдання
1	Реєстрація нового користувача	- Перевірка зрозумілості та зручності форми реєстрації - Оцінка процесу підтвердження реєстрації через електронну пошту.
2	Пошук та бронювання коворкінгу	- Перевірка зручності використання пошукової системи та фільтрів; - Оцінка процесу бронювання, включаючи вибір дати, часу та місця; - Перевірка коректності відображення інформації про бронювання в особистому кабінеті користувача.
3	Перегляд та редагування інформації про коворкінг (для адміністраторів)	- Оцінка зручності навігації в адміністративній панелі; - Перевірка процесу додавання та редагування інформації про коворкінг; - Оцінка відображення статистики та звітів.
4	Залишення відгуків та оцінок	- Перевірка зрозумілості та зручності форми для залишення відгуків; - Оцінка відображення відгуків та оцінок на сторінці коворкінгу.

Повторне тестування з залученням користувачів показало значне покращення зручності використання та загальної задоволеності системою. Більшість користувачів відзначили інтуїтивність навігації, привабливість дизайну та легкість виконання основних завдань.

Тестування користувацького інтерфейсу дозволило переконатися, що система бронювання місць у коворкінгах відповідає очікуванням та потребам користувачів, забезпечуючи комфортний та ефективний досвід взаємодії з системою.

4.3 Висновки за розділом

У четвертому розділі було описано процес тестування розробленого програмного забезпечення для бронювання місць у коворкінгах. Тестування проводилося на трьох рівнях: модульному, інтеграційному та тестуванні користувацького інтерфейсу.

Модульне тестування дозволило перевірити коректність роботи окремих компонентів системи незалежно від інших. Було розроблено набір тестових випадків, які охоплювали ключові функції та методи програмного забезпечення, такі як реєстрація та авторизація користувачів, управління коворкінгами, бронювання місць, відгуки та оцінки, а також адміністративні функції. Результати модульного тестування показали, що більшість компонентів працюють коректно та відповідають заявленим вимогам, а виявлені незначні помилки були оперативно виправлені.

Тестування користувацького інтерфейсу проводилося із залученням групи потенційних користувачів з різним рівнем технічної підготовки. Їм було запропоновано виконати ряд типових завдань, використовуючи вебінтерфейс системи, та поділитися своїми враженнями і зауваженнями. Тестові завдання охоплювали основні сценарії використання системи, такі як реєстрація нового користувача, пошук та бронювання коворкінгу, перегляд та редагування інформації про коворкінг (для адміністраторів), а також залишення відгуків та оцінок. За результатами тестування були внесені правки в дизайн та розташування елементів, покращено зрозумілість інструкцій та підказок, а також оптимізовано деякі процеси взаємодії з системою. Повторне тестування показало значне покращення зручності використання та загальної задоволеності системою.

Проведене тестування дозволило переконатися, що розроблене програмне забезпечення для бронювання місць у коворкінгах функціонує належним чином, відповідає вимогам та очікуванням користувачів, забезпечуючи комфортний та ефективний досвід взаємодії з системою. Виявлені під час тестування недоліки були усунені, що підвищило якість та надійність системи.

ВИСНОВКИ

В атестаційній роботі бакалавра розглянуто актуальну проблему розробки програмного забезпечення для бронювання місць у коворкінгах. Зростання популярності віддаленої роботи та розвиток стартап-екосистеми в Україні зумовлюють потребу в ефективних інструментах для управління гнучкими робочими просторами.

У ході дослідження було проаналізовано сучасний стан ринку послуг коворкінгів в Україні, виявлено основні тенденції та особливості галузі. Аналіз існуючих систем бронювання місць у коворкінгах дозволив визначити ключові функції та особливості, які користувачі очікують від подібних систем.

На основі зібраної інформації було сформульовано функціональні та нефункціональні вимоги до розробки власного програмного забезпечення для бронювання місць у коворкінгах. Ці вимоги стали основою для проєктування архітектури системи, вибору технологій та інструментів для реалізації.

Розроблене програмне забезпечення базується на клієнт-серверній архітектурі з чітким розподілом на фронтенд та бекенд компоненти. Для реалізації використано сучасні вебтехнології, такі як HTML, CSS, JavaScript, PHP, а також системи управління базами даних. Особливу увагу приділено забезпеченню безпеки, продуктивності та масштабованості системи.

Реалізована система включає зручний та інтуїтивно зрозумілий інтерфейс користувача, який дозволяє легко здійснювати пошук, бронювання та управління робочими місцями в коворкінгах. Адміністративна панель надає власникам коворкінгів інструменти для ефективного управління просторами та взаємодії з клієнтами.

Тестування розробленого програмного забезпечення на різних рівнях (модульному та UI) підтвердило його функціональність, надійність та відповідність вимогам. Виявлені недоліки були оперативно усунені, що дозволило підвищити якість та зручність використання системи.

Розроблене програмне забезпечення для бронювання місць у коворкінгах має практичну цінність для власників коворкінгів та їх клієнтів, оскільки дозволяє оптимізувати процеси управління робочими просторами, забезпечує зручність та доступність послуг для користувачів. Впровадження такої системи сприятиме розвитку ринку коворкінгів в Україні та підвищенню якості послуг в цій галузі.

Подальші напрямки розвитку розробленого програмного забезпечення можуть включати інтеграцію з іншими системами та сервісами, розширення функціональності, адаптацію до нових потреб ринку та користувачів. Регулярне оновлення та підтримка системи забезпечать її актуальність та ефективність у довгостроковій перспективі.

В цілому, атестаційна робота бакалавра демонструє успішне вирішення поставленої проблеми розробки програмного забезпечення для бронювання місць у коворкінгах. Отримані результати мають практичну цінність і можуть бути використані для подальшого розвитку та вдосконалення систем управління коворкінгами в Україні.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Wang S.. Coworking space: A new form of urban public space under the influence of sharing economy / S. Wang, G. Ding, T. Zou, D. Liu // *Frontiers of Architectural Research*. — 2021. — Vol. 10, № 2. — P. 288-305.
2. Kojo I., Nenonen S. Coworking space as an enabler of entrepreneurship - Case study in Finland / I. Kojo, S. Nenonen // *ERES 2018: 25th Annual European Real Estate Society Conference*. — 2018. — Vol. 1, № 2. — P. 142-158.
3. Bouncken R. B. The dark side of entrepreneurship in coworking-spaces / R. B. Bouncken, M. M. Aslam, A. J. Reuschl // *Inside the Mind of the Entrepreneur*. — Cham: Springer, 2018. — Vol. 12. — P. 135-147.
4. Orel M. Coworking environments and digital nomadism: Balancing work and leisure whilst on the move / M. Orel // *World Leisure Journal*. — 2019. — Vol. 61, № 3. — P. 215-227.
5. Sankari I. Coworking space as workplace: Characteristics and user experience: master's thesis / I. Sankari. — Aalto: Aalto University, 2019. — 300 p.
6. Bueno S. Coworking spaces: A new way of achieving productivity / S. Bueno, G. Rodríguez-Baltanás, M. D. Gallego // *Journal of Facilities Management*. — 2018. — Vol. 16, № 4. — P. 452-466.
7. Bouncken R. B. How a phenomenon of the sharing economy builds a novel trend for the workplace and for entrepreneurship / R. B. Bouncken, A. J. Reuschl // *Review of Managerial Science*. — 2018. — Vol. 12, № 1. — P. 317-334.
8. Merkel J. 'Freelance isn't free': Co-working as a critical urban practice to cope with informality in creative labour markets / J. Merkel // *Urban Studies*. — 2019. — Vol. 56, № 3. — P. 526-547.
9. Yang E. Coworking space as a third-fourth place: Changing models of a hybrid space in corporate real estate / E. Yang, C. Bisson, B. E. Sanborn // *Journal of Corporate Real Estate*. — 2019. — Vol. 21, № 4. — P. 324-345.

10. Rahimian F. P. Architectural design process: Technology, research, and practice / F. P. Rahimian, J. Goulding, A. M. Salama, S. Ruddock. — Springer Nature, 2021. — 240 p.
11. Repanich J. Why software projects take longer than you think: A statistical model / J. Repanich // ArXiv:1908.06572 [Cs]. — 2019 — Vol. 8, № 1. — P. 310-312.
12. Akbar Z..Effective project management practices for enterprise software development / Z. Akbar, M. Shafiq, T. Kamal, M. Hamza, A. Saba // Proceedings of the 2019 8th International Conference on Educational and Information Technology. — 2019. — Vol. 1, № 1. — P. 175–180.
13. Rischpater R. Web application development with MEAN / R. Rischpater. — Packt Publishing Ltd, 2018. — 128 p.
14. Gupta A., Gupta H. Developing an agile project management framework for an educational institution / A. Gupta, H. Gupta // Journal of Project Management. — 2019. — Vol. 4, № 4. — P. 259-268.

ДОДАТОК А
Текст програми

```

# Файл: api.py
from flask import Flask, request, jsonify
from db import get_db_connection, close_db_connection

app = Flask(__name__)

@app.route('/api/coworkings', methods=['GET'])
def get_coworking_list():
    connection = get_db_connection()
    cursor = connection.cursor(dictionary=True)
    cursor.execute("SELECT * FROM coworkings")
    coworkings = cursor.fetchall()
    close_db_connection(connection)
    return jsonify(coworkings)

@app.route('/api/coworkings', methods=['POST'])
def filter_coworking_list():
    filters = request.json
    query = "SELECT * FROM coworkings WHERE 1=1"
    params = []
    if 'city' in filters:
        query += " AND city = %s"
        params.append(filters['city'])
    connection = get_db_connection()
    cursor = connection.cursor(dictionary=True)
    cursor.execute(query, params)
    coworkings = cursor.fetchall()
    close_db_connection(connection)
    return jsonify(coworkings)

@app.route('/api/coworkings/search', methods=['GET'])
def search_coworking():
    query_param = request.args.get('q')
    connection = get_db_connection()
    cursor = connection.cursor(dictionary=True)
    cursor.execute("SELECT * FROM coworkings WHERE name LIKE
%s", ("% " + query_param + "%",))
    coworkings = cursor.fetchall()
    close_db_connection(connection)
    return jsonify(coworkings)

@app.route('/api/coworkings/<int:coworking_id>', methods=['GET'])
def get_coworking_details
# Файл: api.py

```

```

@app.route('/api/calendar', methods=['GET'])
def load_calendar():
    # Логіка для завантаження календаря
    return jsonify({"message": "Calendar loaded successfully"})

@app.route('/api/coworkings/overview', methods=['GET'])
def load_coworking_overview():
    connection = get_db_connection()
    cursor = connection.cursor(dictionary=True)
    cursor.execute("""
        SELECT c.coworking_id, c.name, c.city, AVG(r.rating) as
average_rating
        FROM coworkings c
        LEFT JOIN reviews r ON c.coworking_id = r.coworking_id
        GROUP BY c.coworking_id
    """)
    overview = cursor.fetchall()
    close_db_connection(connection)
    return jsonify(overview)

@app.route('/api/payments', methods=['POST'])
def process_payment():
    payment_data = request.json
    connection = get_db_connection()
    cursor = connection.cursor()
    cursor.execute("""
        INSERT INTO payments (booking_id, amount, payment_method)
        VALUES (%s, %s, %s)
    """, (payment_data['booking_id'], payment_data['amount'],
payment_data['payment_method']))
    connection.commit()
    close_db_connection(connection)
    return "", 201

if __name__ == '__main__':
    app.run(debug=True)
# Файл: test_api.py
import unittest
import json
from api import app

class APITestCase(unittest.TestCase):
    def setUp(self):
        self.app = app.test_client()
        self.app.testing = True

```

```

def test_get_coworking_list(self):
    response = self.app.get('/api/coworkings')
    self.assertEqual(response.status_code, 200)
    data = json.loads(response.get_data())
    self.assertIsInstance(data, list)

def test_filter_coworking_list(self):
    response = self.app.post('/api/coworkings', json={"city": "Kyiv"})
    self.assertEqual(response.status_code, 200)
    data = json.loads(response.get_data())
    self.assertIsInstance(data, list)

def test_search_coworking(self):
    response = self.app.get('/api/coworkings/search?q=coworking')
    self.assertEqual(response.status_code, 200)
    data = json.loads(response.get_data())
    self.assertIsInstance(data, list)

def test_get_coworking_details(self):
    response = self.app.get('/api/coworkings/1')
    self.assertEqual(response.status_code,

```

```

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Головна сторінка</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="coworking_list.html">Список
коворкінгів</a></li>
                <li><a href="#" id="createEventButton">Створити
подію</a></li>
                <li><input type="text" id="searchBar" placeholder="Пошук
коворкінгу"></li>
            </ul>
        </nav>
    </header>

```

```

    <main>
        <h1>Ласкаво просимо на наш сайт</h1>
        <!-- Інший контент -->
    </main>
    <script src="main.js"></script>
</body>
</html>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Авторизація та реєстрація</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="index.html">Головна</a></li>
            </ul>
        </nav>
    </header>
    <main>
        <h1>Авторизація та реєстрація</h1>
        <div id="authForms">
            <div id="loginForm">
                <h2>Вхід</h2>
                <form id="login">
                    <label for="loginEmail">Email</label>
                    <input type="email" id="loginEmail" name="email">
                    <label for="loginPassword">Пароль</label>
                    <input type="password" id="loginPassword"
name="password">
                    <button type="submit">Увійти</button>
                </form>
            </div>
            <div id="registerForm">
                <h2>Реєстрація</h2>
                <form id="register">
                    <label for="registerName">Ім'я</label>
                    <input type="text" id="registerName" name="name">
                    <label for="registerEmail">Email</label>

```

```

        <input type="email" id="registerEmail" name="email">
        <label for="registerPassword">Пароль</label>
        <input type="password" id="registerPassword"
name="password">
        <label for="registerPhone">Телефон</label>
        <input type="text" id="registerPhone" name="phone">
        <button type="submit">Зареєструватися</button>
    </form>
</div>
</div>
</main>
<script src="auth.js"></script>
</body>
</html>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Управління бронюваннями</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="index.html">Головна</a></li>
            </ul>
        </nav>
    </header>
    <main>
        <h1>Управління бронюваннями</h1>
        <div id="bookingManagement">
            <input type="text" id="bookingSearch" placeholder="Пошук
бронювання">
            <div id="calendar">
                <!-- Календар для бронювань буде завантажений тут -->
            </div>
            <table id="bookingTable">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Ім'я</th>

```

```

        <th>Стіл</th>
        <th>Дата</th>
        <th>Час початку</th>
        <th>Час завершення</th>
        <th>Статус</th>
    </tr>
</thead>
<tbody>
    <!-- Інформація про бронювання буде завантажена тут -->
</tbody>
</table>
</div>
</main>
<script src="booking_management.js"></script>
</body>
</html>

```

```

document.addEventListener('DOMContentLoaded', () => {
    initNavigation();

```

```

document.getElementById('createEventButton').addEventListener('click',
openCreateEventModal);

```

```

=> {
    document.getElementById('searchBar').addEventListener('input', (event)
        searchCoworking(event.target.value);
    });

```

```

    loadCoworkingList();
});

```

```

function initNavigation() {
    // Логіка для ініціалізації меню навігації
}

```

```

function loadCoworkingList() {
    fetch('/api/coworkings')
        .then(response => response.json())
        .then(data => {
            const coworkingList = document.getElementById('coworkingList');
            coworkingList.innerHTML = "";
            data.forEach(coworking => {
                const li = document.createElement('li');
                li.textContent = coworking.name;
            });
        });
}

```

```

        coworkingList.appendChild(li);
    });
}

function openCreateEventModal() {

function searchCoworking(query) {
    fetch(`/api/coworkings/search?q=${query}`)
    .then(response => response.json())
    .then(data => {
        const coworkingList = document.getElementById('coworkingList');
        coworkingList.innerHTML = "";
        data.forEach(coworking => {
            const li = document.createElement('li');
            li.textContent = coworking.name;
            coworkingList.appendChild(li);
        });
    });
}

document.addEventListener('DOMContentLoaded', () => {

document.getElementById('applyFiltersButton').addEventListener('click',
applyFilters);
    loadCoworkingList();
});

function loadCoworkingList() {
    fetch('/api/coworkings')
    .then(response => response.json())
    .then(data => {
        const coworkingList = document.getElementById('coworkingList');
        coworkingList.innerHTML = "";
        data.forEach(coworking => {
            const li = document.createElement('li');
            li.textContent = `${coworking.name} - ${coworking.city}`;
            coworkingList.appendChild(li);
        });
    });
}

function applyFilters() {
    const city = document.getElementById('cityFilter').value;
    fetch('/api/coworkings', {

```

```

        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ city })
    })
    .then(response => response.json())
    .then(data => {
        const coworkingList = document.getElementById('coworkingList');
        coworkingList.innerHTML = "";
        data.forEach(coworking => {
            const li = document.createElement('li');
            li.textContent = `${coworking.name} - ${coworking.city}`;
            coworkingList.appendChild(li);
        });
    });
}

document.addEventListener('DOMContentLoaded', () => {
    const form = document.getElementById('editCoworkingForm');
    form.addEventListener('submit', (event) => {
        event.preventDefault();
        const formData = new FormData(form);
        const data = Object.fromEntries(formData.entries());
        const coworkingId = form.getAttribute('data-coworking-id');
        editCoworking(coworkingId, data);
    });
    loadCoworkingDetails();
});

function loadCoworkingDetails() {
    const coworkingId = new
URLSearchParams(window.location.search).get('id');
    fetch(`/api/coworkings/${coworkingId}`)
        .then(response => response.json())
        .then(data => {
            document.getElementById('name').value = data.name;
            document.getElementById('description').value = data.description;
            document.getElementById('contacts').value = data.contacts;
            document.getElementById('address').value = data.address;
            document.getElementById('city').value = data.city;
            // Фото не заповнюється, так як це поле файлу
        });

    document.getElementById('editCoworkingForm').setAttribute('data-coworking-id',
data.coworking_id);
});
}

```

```
function editCoworking(coworkingId, data) {  
  fetch(`/api/coworkings/${coworkingId}`, {  
    method: 'PUT',  
    headers: { 'Content-Type': 'application/json' },  
    body: JSON.stringify(data)  
  })  
  .then(response => {  
    if (response.ok)
```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Розробка програмного забезпечення бронювання місць у коворкінгу.

Виконав студент групи КНТ-211сп
Керівник проєкту, к.т.н., доцент

Владислав НАДТОЧИЙ
Степан СКРУПСЬКИЙ

Рисунок Б.1 – Слайд 1

Мета та задачі

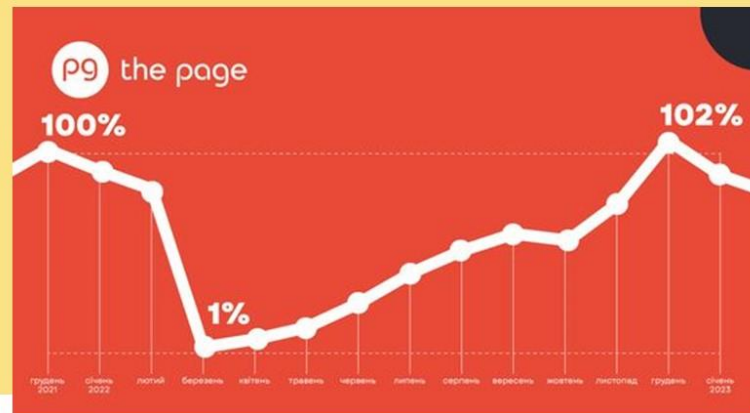
Метою даного дипломного проєкту є розробка програмного забезпечення для бронювання місць у коворкінгах, яке дозволить користувачам легко знаходити та бронювати робочі місця, а власникам коворкінгів - ефективно керувати завантаженістю приміщень.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- аналіз потреб користувачів та вимог до функціоналу системи бронювання місць у коворкінгах;
- дослідження сучасних ІТ-рішень для управління коворкінгами та бронювання місць;
- проектування архітектури програмного забезпечення, що забезпечує масштабованість, безпеку та зручність використання;
- розробка бази даних для зберігання інформації про коворкінги, користувачів та бронювання;
- реалізація функціоналу бронювання місць, управління коворкінгами та оплати послуг;
- тестування розробленого програмного забезпечення.

Рисунок Б.2 – Слайд 2

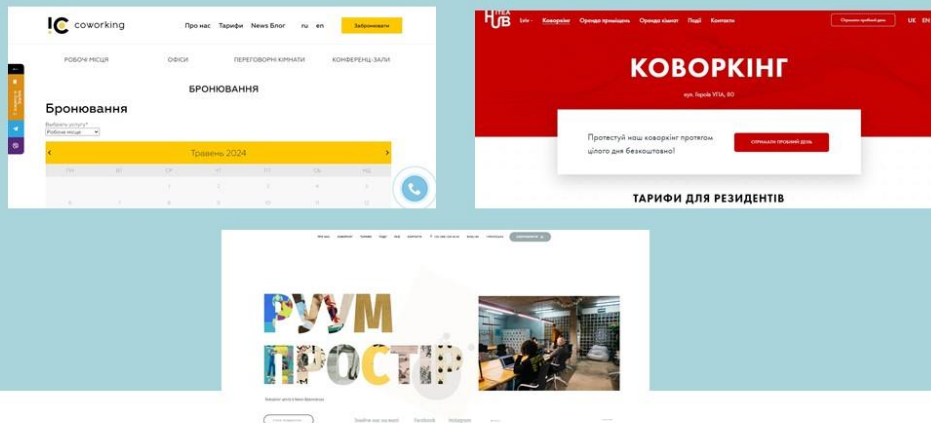
Огляд сучасного стану ринку послуг коворкінгів



3

Рисунок Б.3 – Слайд 3

Аналіз існуючих систем бронювання місць у коворкінгах



4

Рисунок Б.4 – Слайд 4

Порівняння сучасних вебсайтів

Характеристика	IC Coworking	ХАБковоркінг	РушПростір
Основний фокус	Бронювання робочих місць	Інформація про коворкінг та тарифи	Презентація коворкінгу та його спільноти
Дизайн	Мінімалістичний із зручним календарем	Стильний із яскравими візуальними акцентами	Сучасний, креативний дизайн з візуальною привабливістю
Функції	Вибір послуг і дати бронювання	Інформація про приміщення, події, контакти	Інформація про програми, спільноту, блог
Мовні опції	Українська, російська, англійська	Українська, англійська	Українська
Інтерактивні елементи	Кнопка бронювання, календар	Заклик до дії для безкоштовної пробної візиту	Посилання на соціальні мережі, реєстрація/вхід
Соціальні мережі	Контакти через телефон	Посилання на Facebook, Instagram	Посилання на Facebook, Instagram

5

Рисунок Б.5 – Слайд 5

Огляд ключових завдань

Етап	Опис завдань
Визначення потреб користувачів	Аналіз потреб кінцевих користувачів та адміністрації коворкінгів для визначення функціональних вимог до системи.
Аналіз конкурентних рішень	Дослідження існуючих систем бронювання для визначення їх сильних та слабких сторін.
Формулювання вимог до системи	Формулювання технічних та функціональних вимог до нової системи на основі зібраної інформації.
Вибір технологічного стеку	Вибір мов програмування, фреймворків, баз даних та інших технологій для реалізації системи.
Проектування архітектури системи	Створення архітектурної схеми системи, яка описує взаємодію компонентів, модульну структуру та ін.
Розробка інтерфейсу користувача	Дизайн інтерфейсу користувача, який має бути інтуїтивно зрозумілим та зручним.
Розробка основних модулів	Реалізація ключових модулів системи, включаючи модуль бронювання, управління закладом та облік клієнтів.
Тестування	Перевірка системи на наявність помилок, відповідність вимогам і надійність роботи.
Впровадження та налагодження	Запуск системи в експлуатацію з подальшим моніторингом і налагодженням її роботи.
Підтримка та розвиток	Забезпечення оперативної технічної підтримки, оновлення та розширення функціоналу системи.

6

Рисунок Б.6 – Слайд 6

Функції для користувача-клієнта

№	Назва функції	Скорочений опис функції
1	Вхід у систему	Авторизація через електронну пошту та пароль.
2	Реєстрація	Введення особистих даних для створення акаунту.
3	Перегляд коворкінгів	Перегляд інформації про коворкінги без можливості бронювання.
4	Деталі коворкінгу	Показує повну інформацію про коворкінг.
5	Сортування та фільтрація коворкінгів	Фільтрування і сортування за ціною, рейтингом, доступністю місць.
6	Пошук та перегляд схожих коворкінгів	Шукати коворкінги за назвою чи локацією.
7	Бронювання місць	Резервування місця за конкретні дату і час.
8	Управління бронюваннями	Можливість редагувати чи скасовувати бронювання.
9	Оцінювання та відгуки	Залишення відгуків та оцінок після відвідування.

7

Рисунок Б.7 – Слайд 7

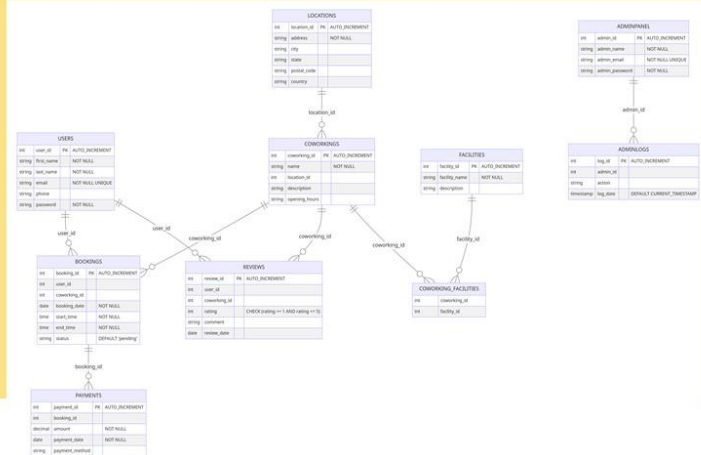
Функції для адміністратора

№	Назва функції	Опис функції
1	Управління інформацією про коворкінг	Додавання, оновлення чи видалення інформації про коворкінги.
2	Перегляд бронювань	Доступ до всіх бронювань коворкінгу для ефективного управління.
3	Створення бронювань персоналом	Внесення в систему бронювань, здійснених через інші канали.
4	Реєстрація нових коворкінгів	Створення облікових записів для нових коворкінгів.

8

Рисунок Б.8 – Слайд 8

Схема бази даних



9

Рисунок Б.9 – Слайд 9

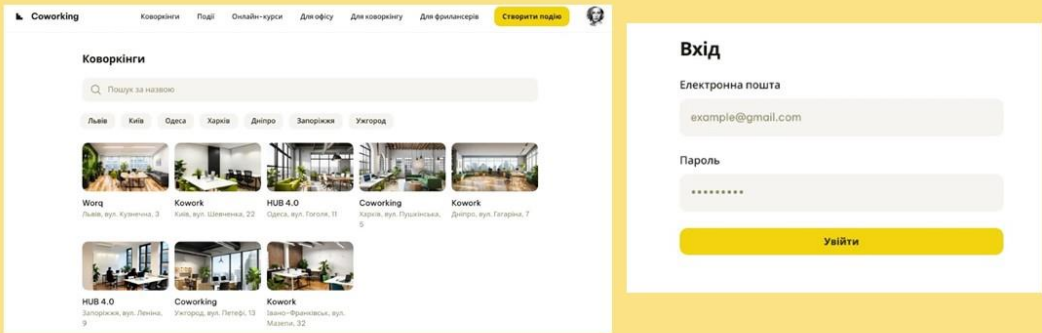
Опис структури вебсайту

Назва файлу	Опис
index.php	Головна сторінка.
coworking_list.php	Список коворкінгів.
coworking_edit.php	Сторінка редагування коворкінгу.
booking.php	Сторінка бронювання.
calendar.php	Сторінка календаря.
coworking_overview.php	Сторінка огляду коворкінгів.
reviews.php	Сторінка відгуків.
coworking_details.php	Сторінка детальної інформації про коворкінг.
auth.php	Сторінка авторизації та реєстрації.
booking_management.php	Сторінка управління бронюваннями.

10

Рисунок Б.10 – Слайд 10

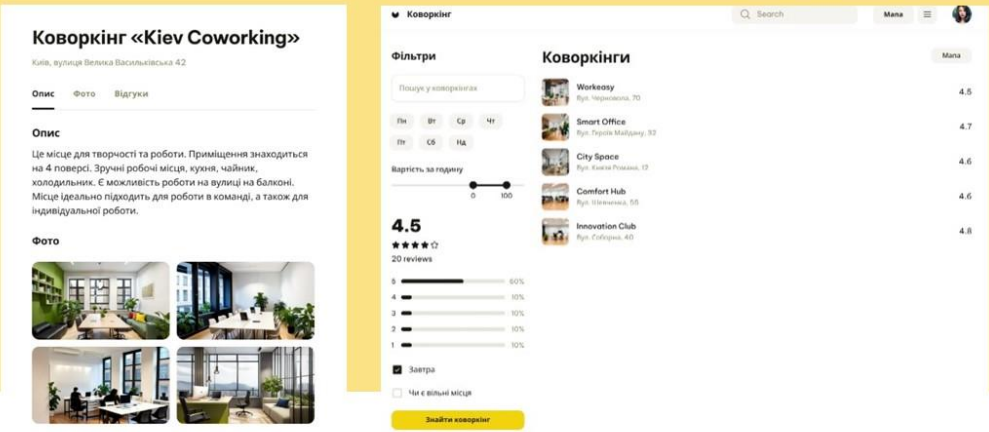
Інтерфейс користувача



11

Рисунок Б.11 – Слайд 11

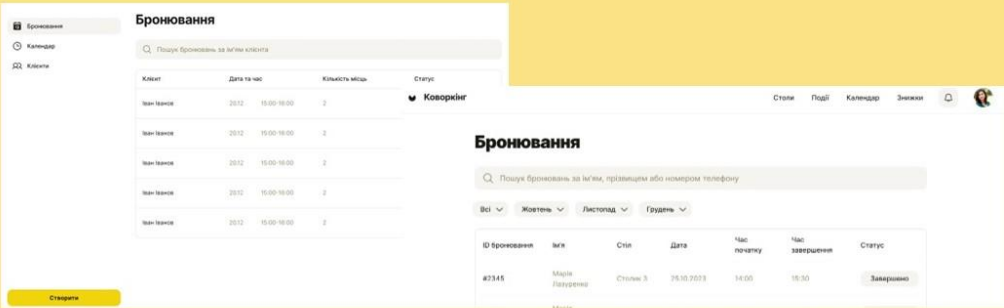
Інтерфейс користувача



12

Рисунок Б.12 – Слайд 12

Інтерфейс користувача



13

Рисунок Б.13 – Слайд 13

Результати модульного тестування

Результати модульного

Компонент	Тестовий випадок	Очікуваний результат	Фактичний результат	Статус
Регістрація користувачів	Регістрація нового користувача з коректними даними	Користувач успішно зареєстрований, створено новий обліковий запис	Користувач успішно зареєстрований, створено новий обліковий запис	Пройдено
	Регістрація з використанням існуючої електронної пошти	Відображення повідомлення про помилку, реєстрація не виконана	Відображення повідомлення про помилку, реєстрація не виконана	Пройдено
Авторизація користувачів	Вхід з використанням коректних облікових даних	Користувач успішно авторизований, перенаправлення на головну сторінку	Користувач успішно авторизований, перенаправлення на головну сторінку	Пройдено
	Вхід з використанням некоректних облікових даних	Відображення повідомлення про помилку, користувач не авторизований	Відображення повідомлення про помилку, користувач не авторизований	Пройдено
Управління коворкінгами	Створення нового коворкінгу з коректними даними	Коворкінг успішно створений, відображається в списку коворкінгів	Коворкінг успішно створений, відображається в списку коворкінгів	Пройдено
	Редагування інформації про існуючий коворкінг	Інформація про коворкінг успішно оновлена	Інформація про коворкінг успішно оновлена	Пройдено
Бронювання місць	Видалення коворкінгу	Коворкінг успішно видалений, не відображається в списку коворкінгів	Коворкінг успішно видалений, не відображається в списку коворкінгів	Пройдено
	Бронювання місця з вибором дати, часу та конкретного місця	Бронювання успішно створене, відображається в списку бронювань користувача	Бронювання успішно створене, відображається в списку бронювань користувача	Пройдено
Відгуки та оцінки	Бронювання місця, яке вже заброньоване	Відображення повідомлення про неможливість бронювання, бронювання не створене	Відображення повідомлення про неможливість бронювання, бронювання не створене	Пройдено
	Додавання відгуку та оцінки для коворкінгу	Відгук та оцінка успішно додані, відображаються на сторінці коворкінгу	Відгук та оцінка успішно додані, відображаються на сторінці коворкінгу	Пройдено
Адміністративні функції	Відображення середньої оцінки для коворкінгу	Середня оцінка коворкінгу розрахована коректно та відображається на сторінці коворкінгу	Середня оцінка коворкінгу розрахована коректно та відображається на сторінці коворкінгу	Пройдено
	Перегляд списку бронювань з можливістю фільтрації та сортування	Список бронювань відображається коректно, фільтрація та сортування працюють належним чином	Список бронювань відображається коректно, фільтрація та сортування працюють належним чином	Пройдено

14

Рисунок Б.14 – Слайд 14



Висновки

Атестаційна робота бакалавра присвячена актуальній проблемі розробки програмного забезпечення для бронювання місць у коворкінгах. У ході дослідження проаналізовано предметну область, сформульовано вимоги до системи та розроблено клієнт-серверне програмне забезпечення з використанням сучасних вебтехнологій. Реалізована система має зручний інтерфейс користувача та адміністративну панель для ефективного управління коворкінгами. Тестування підтвердило функціональність та надійність розробленого продукту. Отримані результати мають практичну цінність для розвитку ринку коворкінгів в Україні та можуть бути використані для подальшого вдосконалення систем управління гнучкими робочими просторами.