

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ
СПІЛЬНОЇ ОРЕНДИ ЖИТЛА. RESEARCH AND SOFTWARE
IMPLEMENTATION OF THE APPLICATION OF JOINT HOUSE RENTAL

Виконав: студент(ка) 2 курсу, групи КНТ-211м
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

Вавілов М.О.

(прізвище та ініціали)

Керівник Зайко Т.А.

(прізвище та ініціали)

Рецензент Скрупський С.Ю.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 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.
С.О. Субботін
 “ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА(КИ)

Вавілова Миколи Олександровича

(прізвище, ім'я, по батькові)

- Тема проекту (роботи) Дослідження та програмна реалізація застосунку спільної оренди житла.
Research and Software Implementation of the Application of Joint House Rental
 керівник проекту (роботи) Зайко Тетяна Анатоліївна,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
 затверджені наказом закладу вищої освіти від “4” листопада 2022 року № 363
- Строк подання студентом проекту (роботи) 30 листопада 2022 року
- Вихідні дані до проекту (роботи) рекомендована література, технічне завдання
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка архітектури програми. 3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація, тестування та експериментальне дослідження програми.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Зайко Т.А., доцент		
Нормоконтроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання 12 жовтня 2022 р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Постановка завдання роботи.	1-2 тижні	Завдання, ТЗ
2	Аналіз предметної області.	3-5 тижні	Розділ 1
3	Вибір мови програмування та інших технологій розробки	6-7 тижні	Розділ 2
4	Розробка архітектури програми.	8–11 тижні	Розділ 3
5	Розробка програми.	12 тиждень	Розділ 4
6	Тестування та експериментальне дослідження програмного забезпечення	13 тиждень	
7	Оформлення пояснювальної записки та документів до неї	14 тиждень	Додатки
8	Нормоконтроль та рецензування.	15 тиждень	
9	Захист роботи.	16 тиждень	

Студент (ка) Вавілов М.О.
(підпис) (прізвище та ініціали)Керівник проекту (роботи) Зайко Т.А.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра: 93 с., 6 табл., 46 рис., 3 дод., 11 джерел.

БРОНЮВАННЯ ЖИТЛА, ВЕБЗАСТОСУНОК, JAVASCRIPT, VUE.JS, PHP.

Об'єкт дослідження – процес розробки та аналізу програмного забезпечення для бронювання житла.

Предмет дослідження – Програмні засоби для вебзастосунку бронювання житла.

Метою роботи є дослідження та розробка вебзастосунку для бронювання житла.

Матеріали, методи та технічні засоби: С-подібна скриптова мова PHP, спеціальна мова стилів CSS, мова розмітки гіпертексту HTML, фреймворки Laravel та Vue.js, бібліотека UI-компонентів для Vue, персональний комп'ютер з процесором Intel Core i5-9400f під управлінням операційної системи Microsoft Windows 10.

Результати. Створено вебзастосунок для букінгу номерів готелів або хостелів за допомогою фреймворків Laravel та Vue.js з бібліотекою компонентів Vuetify.

Висновки. Розроблено онлайн сервіс для бронювання житла з простим й прозорим інтерфейсом, зрозумілий дизайн для користувача й гарним функціоналом який може конкурувати з аналогами.

Галузь використання – будь який ПК або сучасний смартфон з підтримкою браузера.

ABSTRACT

Explanatory note to the diploma qualification work of the master: 93 pages, 6 tables, 3 appendixes, 11 sources.

ACCOMMODATION RESERVATION, WEB APPLICATION, JAVASCRIPT, VUE.JS, PHP.

The object of research is the process of development and analysis of software for housing reservations.

The subject of research is web applications for housing reservations.

The purpose of this work is research and development of a web application for housing reservations.

Materials, methods, and tools: C-like scripting language PHP, special styling language CSS, hypertext markup language HTML, Laravel and Vue.js frameworks, UI-component library for Vue, personal computer with Intel core i5-9400f processor running Microsoft Windows 10 operating system.

Results. A web application for booking hotel or hostel rooms was created using Laravel and Vue.js frameworks with the Vuetify component library.

Conclusions An online service for housing reservations has been developed with a simple and transparent interface, clear desing for the user and good function that can complete with analogues.

The field of use is any PC or smart phone with browser support.

ЗМІСТ

	С.
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Вебзастосунок і його типи.....	10
1.2 Аналіз існуючих аналогів.....	11
1.3 Загальна схема системи.....	18
2 Розробка архітектури програми.....	20
2.1 Необхідні технології для створення вебзастосунку.....	20
2.2 Обґрунтування вибору front-end фреймворку.....	23
2.3 Обґрунтування вибору back-end фреймворку.....	27
2.4 Вибір фреймворків.....	29
2.5 Редактор коду Visual Studio Code.....	30
2.6 Вибір системи управління базою даних.....	31
2.7 Діаграма взаємодії клієнта і сервера.....	32
2.8 Діаграма дій користувача у системі.....	34
2.9 Розробка дизайну сайту.....	35
3 Основні рішення щодо реалізації компонентів системи.....	37
3.1 Опис структури файлів проєкту.....	37
3.2 База даних.....	39
3.3 Логіка обробки HTTP-запитів та її шаблони.....	44
4 Експлуатація та тестування програми.....	46
4.1 Призначення та умови застосування вебзастосунку.....	46
4.2 Налаштування та підключення за допомогою OpenServer.....	47
4.3 Робота з вебзастосунком.....	52
4.4 Можливі модифікації.....	56
4.5 Вхідні та вихідні дані.....	56
Висновки.....	57
Перелік джерел посилань.....	58
Додаток А Технічне завдання.....	59

Додаток Б Опис програми.....	63
Додаток В Текст програми.....	66
Додаток Г Слайди презентації.....	81

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CRUD – Create Read Update Delete;

DOM – Document Object Model;

MPA – Multi Page Application;

PWA – Progressive Web App,

SaaS – Software as a Service;

SPA – Single Page Application;

СУБД – Система Управління Базами Даних.

ВСТУП

Необхідність в оренді житлової площі є досить розповсюдженою і з цією задачею справляються вебсервіси з оренди житла. Мова йде про готелі, квартири, будинки, хостели тощо. Адаже при подорожах, люди хочуть знайти місце для ночівлі в будь-якому куточку світу.

Через великий вибір вебзастосунків не завжди легко обрати яким користуватися, адже різниця між ними і якість в обслуговуванні відрізняється з кожним новим сервісом який користувач знаходить в пошуках оренди.

Окрім мандрівників, власники готелів, хостелів чи будь-якої орендованої житлової площі також приділяють особливу увагу вебдодаткам які є популярними для подорожей по світу та оренди, бо велика частина клієнтів зазвичай надходить саме з цих платформ онлайн-бронювання.

Основною метою завдання дипломної роботи є реалізація вебдодатку для оренди житла, який буде використовуватися в певному місті і буде надавати змогу легко і швидко орендувати необхідну житлову площу в декілька кліків і буде конкурентноспроможним в сучасних реаліях.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Вебзастосунок і його типи

Вебзастосунок (Вебдодаток) – це програма або програмне забезпечення, яке можна відкрити будь-яким сучасним браузером. Інтерфейс програми створюється за допомогою таких мов програмування як: HTML, CSS, Javascript, які підтримуються в будь-якому сучасному браузері (Opera, Chrome, Mozilla). Серверна частина програми (Back-end) використовує для написання такі мови: Python, PHP, Ruby, Java, з відповідними фрейворками.

Основні переваги вебзастосунків:

- застосовуються на будь-якій операційній системі, оскільки підтримують усі сучасні браузери;
- у зв'язку з тим, що використовується той самий код, що й в desktop додатках їх набагато легше підтримувати;
- простіше програмувати оскільки він не включає багато роботи з елементами системи ПК(ядро, процесор, відеокарта);
- перевага над мобільними додатками в тому, що немає необхідності в схваленні жлдних платформ, для випуску застосунка;
- вебдодатки є більш економним варіантом для будь-якого підприємства оскільки вебпрограми не вимагають підписки або покупки ліцензій, а можуть використовуватися як SaaS-сервіс, що значно дешевше.

Види вебдодатків:

- SPA – односторінковий інтерактивний додаток. Важливо, що він не тільки знаходиться на одній сторінці, а й, подібно до повноцінної програми, є інтерактивним. Так інформаційний вебсайт може складатися з однієї сторінки, але, по суті, не бути SPA. В односторінковому вебдодатку користувач, перемикаючись між вкладками, залишається на одній сторінці, не завантажуючи частини контенту, котрі на використовуються в даний момент, що підвищує швидкість SPA. Гарний приклад SPA є Gmail, можна одразу побачити відмінну ознаку односторінкового додатку, коли користувач

перемикається між списками повідомлень, адреса сторінки не змінюється. Основна мова для створення цього типу застосунків є JavaScript. Простий додаток можна зробити за допомогою бібліотеки JQuery, однак більш оптимальним варіантом для середніх та великих проєктів буде використання фрейворків Vue, Angular або React;

- МРА – багатосторінкова вебпрограма. При взаємодії користувача з вебсайтом, відбувається завантаження нових HTTP-сторінок, через що обмін даними відбувається повільніше ніж у односторінковому додатку. Гарним прикладом багатосторінкового застосунку є інтернет-магазини, такі як Rozetka та Amazon;

- PWA – прогресивний вебдодаток найбільш приближений за своїм функціоналом та якістю досвіду користувача до нативних комп'ютерних та мобільних додатків. Чіткої різниці між PWA додатком та не-PWA немає, можна лише виділити низку характеристик. Прогресивний додаток для запуску має містити проксі-шар та Web App маніфест, де браузер буде виступати віртуальною машиною для запуску цього додатку, подібно до запуску exe-файлу на Windows, або apk на Android. Service Worker це проксі-шар між серверною та клієнтською частиною через яких проходять усі запити, пишеться PWA зазвичай на JS.

1.2 Аналіз існуючих аналогів

1.2.1 Опис вебзастосунку Booking.com

Booking.com – це система бронювання житла, найвідоміша та найстаріша, має підтримку 43 робочих мов, відсутність обов'язкової реєстрації для резервування номеру[1].

Сервіс booking зацікавлює користувача простим та зрозумілим інтерфейсом, можливістю подивитися різні варіанти розміщення з великою кількістю фотографій й відібрати відповідну пропозицію яка найбільш

зацікавила дивлячись на користувальницьку оцінку з реальними відгуками інших туристів (рис. 1.1).

Найбільш примітна перевага цього сервісу є величезна база готелів, хостелів, гостьових будинків з великим ціновим різноманіттям, як для бюджетного відпочинку, так і для забезпечених людей. При цьому є можливість бронювати кілька житлових приміщень для страховки, а перед від'їздом безкоштовно скасовувати резерв.

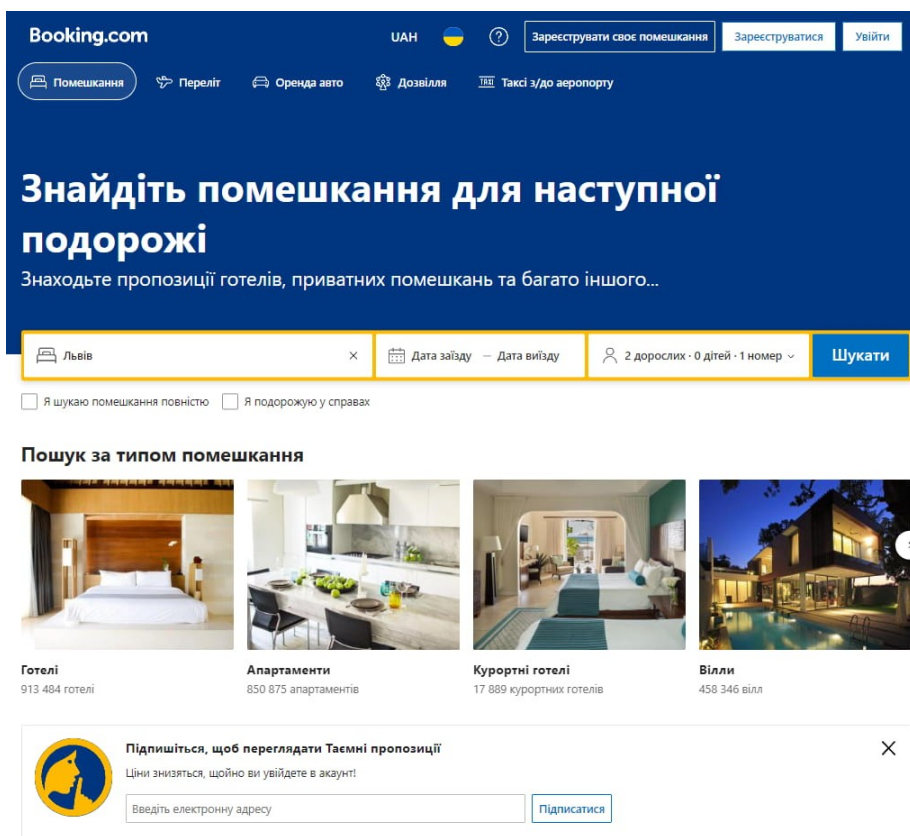


Рисунок 1.1 – Інтерфейс вебдодатку Booking

Переваги цього сервісу:

- простий та зрозумілий інтерфейс;
- безкоштовне скасування броні;
- резервування без внесення оплати;
- безліч пропозицій будь-якого по всьому світу;
- ексклюзивні і спеціальні пропозиції;
- правдивість відгуків користувачів;

- можливість орендувати тут же машину і купити авіаквитки;
- пропозиції в більшості країн світу.

Недоліки цього сервісу:

- не має захисту від фрода;
- іноді є накрутки на ціни.

1.2.2 Опис вебзастосунку Agoda.com

Agoda.com – цей сайт зазвичай входить в рейтинги сервісів бронювання готелів через відкриту цінову політики, без прихованих комісій, інформація про варіанти розміщення не вводить в оману, є правдивою. До корисної інформації яку надає цей сервіс є описи курортів та районів різних міст в котрих користувач збирається бронювати номер, фільтр по необхідності вакцинації від ковід та карантину (рис. 1.2).

Для зареєстрованих користувачів періодично є гідні знижки які можуть досягати 30%. Далеко не в кожному сервісі є зручний розподіл по країнам, містам і навіть районам. В інших деталях не сильно відрізняється від конкурентів, практичний, є безкоштовні додатки, підтримка в чаті, обратний зв'язок.

Невелика хитрість полягає в тому що ціни вказані на сайті без урахування податків.

Через інцидент в 2018 році з крадіжкою майна одного клієнту, відгук котрого сервіс блокував, було виявлено що Agoda штучно розбавляє негативні відгуки позитивними, аби готелі не мали низького рейтингу в їх базі.

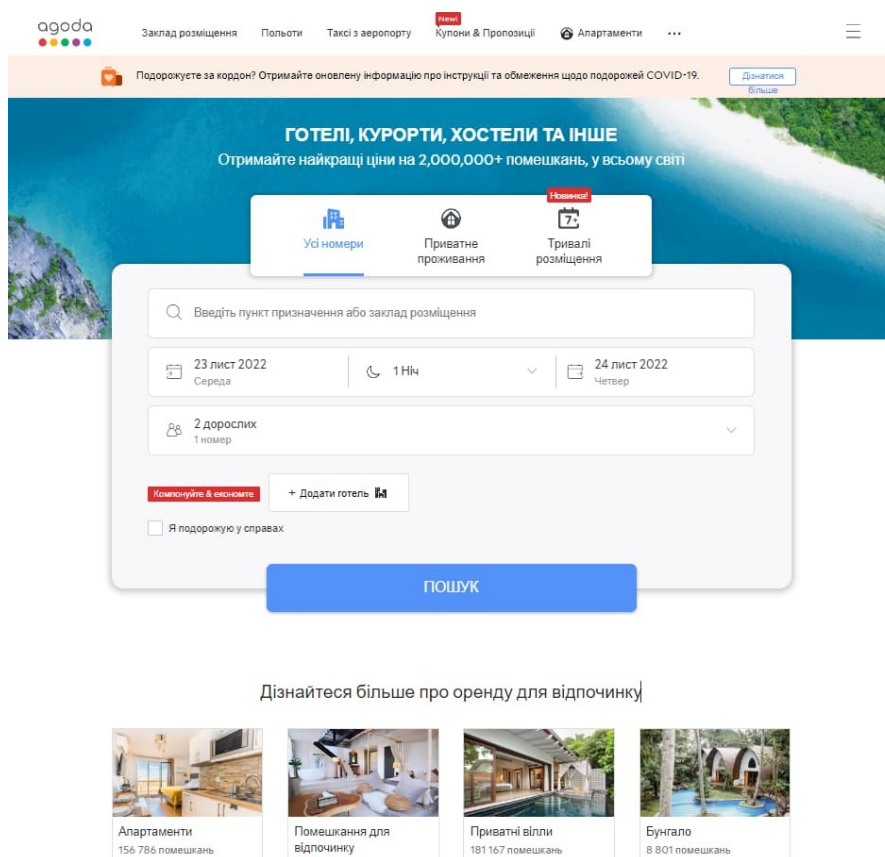


Рисунок 1.2 – Інтерфейс вебдодатку Agoda

Переваги цього сервісу:

- простий та швидкий пошук по сайту;
- бонусна система;
- прозорість умов бронювання;
- гнучкість цінової політики.

Недоліки цього сервісу:

- зайві витрати при оплаті іншою валютою через конвертацію;
- Податки на бронь можна подивитися лише в особистому кабінеті[2].

1.2.3 Опис вебзастосунку Expedia.com

Expedia.com – в топи по сервісам бронювання готелів часто потрапляє цей англomовний ресурс, має одні з найнижчих цін на ринку.

Був розроблений ще у 1996 році як одна з перших туристичних онлайн-агенцій, завдяки більш ніж 70+ терабайтам даних і 20+ рокам інновацій є однією з найбільших у світі туристичних платформ.

Роботаючи вже 2 десятиліття надійно й без перебоїв система не має жодних проблем ні з клієнтською частиною ні з технічною. Надає можливість внесення коштів при заселенні в номер, чого не можуть собі дозволити російські сервіси. Є можливість відшкодувати різницю між ціною цього сайту з конкурентами й отримати купон на знижку в 50 відсотків, але при умові що більш низька ціна була знайдена на американській системі букінгу, а пропозиції повинні відповідати по датах заселення та категоріям номерів.

Так само як і більшість аналогів в цій сфері надає можливість купувати авіаквитки і орендувати автомобілі по всьому світу. Пошук варіантів які підійдуть будь-якому клієнту здійснюється досить швидко порівнянно з тією ж Агод. Процедура оформлення абсолютно прозора і проста, в кілька кліків (рис. 1.3).

The image shows the top section of the Expedia website. At the top, there is the Expedia logo and a 'More travel' dropdown menu. Below this is a navigation bar with tabs for 'Stays', 'Flights', 'Cars', 'Packages', 'Things to do', and 'Cruises'. The 'Stays' tab is selected. The main search area contains a 'Going to' field, a 'Check-in' field set to 'Nov 30', and a 'Check-out' field set to 'Dec 1'. There are also checkboxes for 'Add a flight' and 'Add a car'. A blue 'Search' button is positioned below these fields. To the right of the search area, it says '1 room, 2 travelers'. Below the search area is a large banner for a 'Black Friday sale: Members save 30%'. The banner includes the text 'Collect memories not things. With our member-only sale, save on select hotels through November 28.' and a 'Shop member deals' button. At the bottom of the banner, there is a small disclaimer: '*Book by 11:59 11/28/22 in hotel property local timezone, travel by 6/30/23. Terms and conditions apply. See site for details.' and a 'Help' button.

Рисунок 1.3 – Інтерфейс вебдодатку Expedia

Переваги цього сервісу:

- наявність як недорогих так і елітних готелів;
- не має необхідності пред'явлення розрукованої броні для заселення в номер;
- вартість навіть нижче, ніж у самих готелів, є преференції та бонуси для зареєстрованих користувачів;
- попередження про небезпеку подорожування по території України в воєнний час;
- практичність фільтрів.

Недоліки цього сервісу:

- немає української мови інтерфейсу[2].

1.2.4 Опис вебзастосунку Hotellook.com

Проста і швидка система пошуку та порівняння цін для проживання в готелях охоплюючи найбільш популярні сервіси такі як Agod, Hotels, Booking, Expedia і інші. Серед аналогів має найбільшу базу пошуку, аналізую понад 700 000 хостелів, готелів, гостьових будинків по всьому світу видаючи знижки до 60%.

Цей вебзастосунок є більш оптимальним варіантом аніж користуватися якимсь одним сервісом резервування житла для тих хто хоче заощадити. Дизайн сайту з меню зібрано компактно и максимально зручно. Завжди є свіжі відгуки з фото об'єктів оренди.

Фільтрування кращих пропозицій для резервування здійснюється тут за допомогою спеціального фільтра, до якого входить вибір валюти з більш ніж двадцяти варіантів. На видачу даних з пошуку по об'єктам які зацікавили користувача йду лише пара секунд, навпроти пропозицій вказано рейтинг, що допомагає прийняти правильне рішення (рис. 1.4).

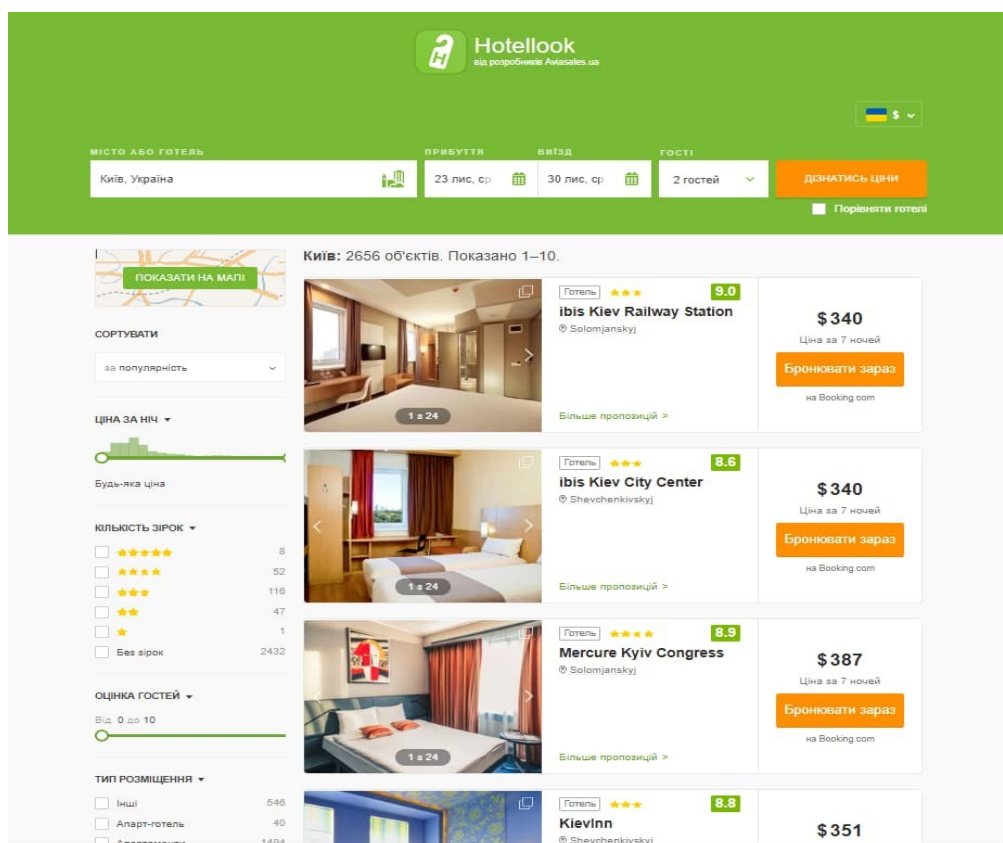


Рисунок 1.4 – Інтерфейс вебдодатку Hotellook

Переваги цього сервісу:

- швидкість завантаження і роботи;
- кількість партнерів;
- найбільша вибірка цін в різних сервісах;
- підтримка більшості мов;
- фільтрація по валюті, типу розміщення, послуг які входять до місця проживання та багато інших параметрів.

Недоліки цього сервісу:

- автоматична переадресація на сайти партнерів по бронюванню номерів[2].

1.3 Загальна схема системи

Після проведення дослідження аналогів вебзастосунків для оренди житлової площі, було визначено вимогу до функціоналу вебдодатку. Технічне завдання наведено у додатку А.

Загальна схема роботи користувача із вебзастосунком.

Користувач якому необхідно орендувати житло переходить до вебзастосунку. Перше що побачить користувач буде головна сторінка, на якій будуть найбільш популярні пропозиції оренди цього вебзастосунку, навігаційна панель на якій будуть посилання на інформацію сторінку про компанію, відгуки, кнопки для реєстрації та авторизації. При бронюванні номеру готелю одразу буде запропоновано зареєструватися для подальшого бронювання, щоб дані броні користувача збереглися в його профіль.

Основний функціонал вебдодатку має вміщувати в себе:

- головну сторінку;
- сторінка для реєстрації та авторизації;
- сторінка з інформацією профілю акаунта;
- сторінка з інформацією про компанію;
- вкладка відгуків користувачів.

Головна сторінка буде вміщувати в себе основну навігацію по застосунку та можливість обрати житло для оренди.

Сторінка реєстрації та авторизації буде містити в собі поля вводу даних користувача при реєстрації та авторизації у вебзастосунку, з можливістю відправити собі на пошту свій пароль від акаунту в разі якщо користувач його забув.

Сторінка інформації профілю акаунта буде вміщувати в собі основну інформацію про користувача та в разі бронювання інформація про стан броні.

Сторінка з інформацією про коомпанію буде вміщувати коротку інформацію про автора вебзастосунку та його мету створення цього застосунку.

Вкладка відгуків користувачів буде містити відгуки тих, хто вже користувався послугами цього застосунку та залишив відгук про якість послуг.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

2.1 Необхідні технології для створення вебзастосунку

2.1.1 HTML

HTML (HyperText Markup Language – в перекладі з англійської мова гіпертекстової розмітки) – мова яка дозволяє описувати структуру вебсторінок документів, формувати абзаци, списки, заголовки та інші структури з звичайного тексту, створювати посилання на інші сторінки. В цій текстовій мові, інструкції з форматування, що називаються тегами, вбудовані в розділи сторінки, яка містить конкретну інформацію. Теги необхідні для повідомлення браузеру, як формувати і виводити інформацію на екран.

У 1989 році Бернерс-Лі запропонував впровадити на базі Internet гіпертекстову систему документів, а вже наприкінці 1990-х років він розробив HTML і написав програмне забезпечення браузера та сервера для запропонованої системи. Того ж року інженери інформаційних систем CERN Тім Бернерс-Лі та Роберт Кайлот спільно подали заявку на фінансування проекту, але проект офіційно не був прийнятий CERN.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML зображення та інші об'єкти, такі як інтерактивні форми, можна вставляти у відтворені сторінки. HTML надає можливість створювати структуровані документи шляхом позначення структурної семантики тексту, наприклад заголовків, абзаців, списків, посилань, цитат та інших елементів. Елементи HTML розділені тегами, написаними за допомогою кутових дужок. Такі теги, як `<img />` або `<input />`, відображають вміст безпосередньо на сторінці. Інші теги, такі як `<p>`, оточують текст і надають інформацію про нього, і можуть містити інші теги як дочірні елементи. Браузери не відображають теги HTML, але використовують їх для інтерпретації вмісту сторінки.

HTML може вбудовувати програми, написані мовами сценаріїв, такими як JavaScript, які впливають на поведінку та вміст веб-сторінок. Включення CSS визначає зовнішній вигляд і макет вмісту. Підтримує стандарти World Wide Web Consortium (W3C) для HTML і CSS[3].

2.1.2 CSS

Це каскадні таблиці стилів, або, перефразовуючи більш доступною мовою, CSS — це технологія, яка описує зовнішній вигляд сторінки, написаної на HTML.

Основне завдання таблиці стилів — відокремити код сторінки від її зовнішнього вигляду. Можливо, ви запитаете: «Чому ви хочете це зробити?». Відповідь очевидна: «Для більшої мобільності». Насправді, припустимо, у вас є сайт на 100 сторінок (це небагато). Припустимо, шрифт для звичайного тексту на вашому веб-сайті становить 15 пунктів. Уявіть, що б ви зробили, якби ви хотіли зробити не 15pt, а 17pt?

Якби ви не використовували CSS, вам довелося б змінювати розмір тексту з 15 pt на 17 pt у кожному місці, де ви вставляєте текст, на кожній зі 100 сторінок. Звичайно, це займе години.

HTML може вбудовувати програми, написані мовами сценаріїв, такими як JavaScript, які впливають на поведінку та вміст веб-сторінок. Включення CSS визначає зовнішній вигляд і макет вмісту. Підтримує стандарти World Wide Web Consortium (W3C) для HTML і CSS.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже

наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо[4].

2.1.3 PHP

Це скриптова мова програмування, для генерації HTML-сторінок на сторони вебсервера, є однією з найпоширеніших мов, що використовуються при створенні вебсайтів, вебзастосунків тощо[5]. У великих проектах використовується іноді спільно з Java, Python, Ruby для отримання більш широкого спектру можливостей, для об'єднання спеціалістів з різними знаннями в одну групу для швидкого вирішення поставленої задачі.

Завдяки тому що мова досить легка для вивчення, вона користується популярністю як серед програмістів-новачків, так і досить досвідченими розробниками вебдодатків. Новачки захоплюються широким функціоналом, легкими у застосуванні можливостями, та великою базою користувачів які майже завжди можуть допогти порадою чи посиланням на ресурс з необхідною інформацією. Більш тямущі користувачі вже зважають увагу на бібліотеки для даної мови й якість способів подання тої чи іншої задачі.

Основні переваги цієї мови:

- легкість і доступність налаштування;
- наявність інтерфейсів до баз даних;
- легко поєднується з іншими мовами;
- підтримує більшість баз даних;
- швидко оброблюються сценарії;
- часті оновлення.

Невеликі недоліки цієї мови:

- непередбачувані зміни в нових версіях;
- деякі функції мають досить складні назви.

2.1.4 JavaScript

Мова програмування з найбільшою динамікою популярності за останні роки, на українському ринку розробників має лідуючу позицію в порівнянні з іншими мовами в комерційному використанні, як додаткова мова є беззаперечним лідером.

Це об'єктно-орієнтована клієнтська мова, яка має величезну кількість додатків для роботи з дизайном, та ще більшу кількість бібліотек які постійно оновлюються та доповнюються. Вона дозволяє реалізувати рішення ряду складних задач в вебдокументах, дозволяє робити сторінки вебсайту інтерактивними для користувачів.

При побудові проекту вебзастосунку при визначенні мови для поставленої задачі, є основа у вигляді HTML та CSS, але далі блок третім блоком для розробки сайту буде JavaScript, на основі котрого будується більшість стандартних вебінтерфейсів.

Ядро JS налічує в собі ряд функцій, які надають можливості:

- зберігати дані в змінних;
- відповідно до певних сценаріїв, здійснювати активацію частини коду на сторінці сайту;
- розміщувати контент який буде сам оновлюватися;
- повна підтримка управління мультимедійними можливостями.

Також ще більш функціоналу можна отримати з допомогою такої надбудови як інтерфейс прикладного програмування (API).

2.2 Обґрунтування вибору front-end фреймворку

2.2.1 Vue.js

Це front-end JS фреймворк з відкритим кодом створений Еваном Ю, випущений ще в 2013 році. Є одним з найпростіших в роботі фреймворком який ліквідує проблеми з якими частіше за все стикаються користувачі Angular.

Основні переваги цього фреймворку:

- існує можливість працювати з visual-DOM;
- можливості DOM-дерева можна розширити за допомогою додаткових директив;
- невеликий розмір;
- низький поріг входу для розробників
- велика кількість додаткових готових бібліотек.

Ключова особливість цього фреймворку в можливості праці з вже існуючим DOM-деревом. Якщо у розробника вже є HTML-розмітка, яку повертає сервер, то він може за допомогою компілятора Vue.js її використовувати.

Користувальницькі директиви розширення можливостей DOM-дерева надають зручний інтерфейс для низькорівневої взаємодії з елементами дерева. Організація прослуховування та обробки подій, або реалізувати невеликі зміни дерева, яєі не вкладаються в парадигму окремого компонента можливі саме завдяки цим директивам.

Розмір фреймворка найменший серед найближених до нього (Angular, React), навіть старий JQuery буде займати більше місця.

Додаткові бібліотеки є для будь-якої ситуації, якщо необхідно зручне управління станом застосунку, є готове рішення у вигляді бібліотеки Vuex, офіційний роутер Vue Router, набір інструментів для розробників Vue CLI, а також Vue Devtools для прискореної отладки застосунку.

2.2.2 React

Цей front-end JS фреймворк з відкритим кодом створений Джорданом Валке, розробником програмного забезпечення у Facebook. Згідно до опитування проведеного Stack Overflow Developer, цей frontend фреймворк є найпопулярнішим у 2021 році.

React спрощує створення інтерактивних інтерфейсів, дозволяючи збирати складний інтерфейс з маленьких блоків ізольованого коду.

Виділяється своєю віртуальною об'єктною моделлю документа DOM, як і Vue.js, з винятковою функціональністю. Даний фреймворк є непоганим варіантом для тих, хто очікую високий трафік і потребує стабільних платформи для обробки цього самого трафіку.

Команда розробників цього фреймворку забезпечила зручний інтерфейс користувача, для більшої ремонтоздатності коду, через проблеми з постійним додаванням функцій у додаток.

Основні переваги цього фреймворку:

- економія часу при повторному використанні компонентів;
- полегшений досвід користувача завдяки DOM;
- альтернатива написання компонентів у React за допомогою hooks, дозволяє писати компоненти без класів поліпшуючи вивчення React.

Основні недоліки цього фреймворку:

- через чисельні і часті оновлення відсутня документація;
- порівнянно довгий період вивчення;
- складнощі вивчення розширення синтаксису JSX.

2.2.3 Angular

Фреймворк один з небагатьох заснований на TypeScript, офіційно запущений у 2016 розроблений Google, з метою подолати складнощі у вимогах постійно зростаючих технологій і звичайними концепціями.

Відмінність від React полягає у функції двохстороннього зв'язування даних, що являє собою синхронізацію між моделлю та представленням в реальному часі, де будь-яка зміна у моделі буде миттєво відображатися на відображенні даних для користувача та навпаки.

Основні переваги цього фреймворку:

- вбудована функціональність для контролю та оновленню змін, внесених у модель та виду даних для користувача в обох панелях;
- зменшений обсяг коду, оскільки більшість відомих функцій і двустороннє зв'язування даних, надано за замовчуванням;
- відокремлення компонентів від залежностей, визначаючи їх зовнішні елементи;
- повторне використання компонентів можливе, ними легко користуватися завдяки ін'єкції залежностей;
- велика спільнота для навчання та допомоги.

Основні недоліки цього фреймворку:

- сам по собі досить складний;
- Angular для деяких проектів просто не потрібен, можна обійтись і більш простими рішеннями;
- Складне перенесення застарілих схем з AngularJS на Angular.

2.2.4 JQuery

Одна з найперших JavaScript бібліотек з відкритим кодом представлених ще у 2006 році. Незважаючи на свій вік досить є вкрай популярним серед розробників, не тільки через легке використання, а й за зменшення потреби написання широкого коду JavaScript.

JQuery використовується для роботи з DOM і CSS на фронтенду та розширює можливості інтерактивності та функціональності вебсайту, синтаксис розроблений щоб зробити орієнтування у навігації зручнішим завдяки вибору DOM-елементів, створенню анімації, обробки подій та розробки AJAX-застосунків[6].

Для створення мобільних застосунків використовувати необхідно JQuery Mobile, оскільки просто JQuery не призначений для цього. Mobile версія з останніми оновленнями дозволяє розробникам створювати власні мобільні програми зі схемою інтерфейсу користувача на основі HTML5-JQuery Mobile.

Основні переваги цього фреймворку:

- спрощена передача HTTP-запитів;
- гнучка модель DOM для накопичення або видалення компонентів;
- HTTP-запити спрощено;
- спрощений динамічний контент.

Основні недоліки цього фреймворку:

- відносно повільна працездатність;
- велика кількість більш гнучких та розширених альтернатив;
- API об'єктної моделі документа застарів[7].

2.3 Обґрунтування вибору back-end фреймворку

2.3.1 Laravel

Laravel – це бекенд фреймворк PHP із відкритим вихідним кодом для розробки вебзастосунків на основі Symfony, які дотримуються архітектури модель–подання–контролер (MVC)[8].

Має вбудовану модульну пакувальну систему, оснащену спеціальним менеджером залежностей. Один з найкращих фреймворків для роботи з бекендом.

Laravel також надає кілька способів доступу до баз даних, а також утиліти для обслуговування та розгортання застосунків.

Основні переваги цього фреймворку:

- досить проста реалізація аутенфікації з легкою конфігурацією логіки авторизації та легкий контроль ресурсів доступу;
- простий API, який гарно працює з бібліотекою SwiftMailer для пошти в PHP, з підтримкою надсилань сповіщень через кілька каналів доставки;
- підтримка Redis, Memcached та інших систем кешу, можливість налаштовувати кілька конфігурацій кешу;

- проекти на Laravel мають попередньо налаштовану обробку винятків і помилок, також має інтеграцію бібліотеки журналювання Monolog і пропонує підтримку декількох обробників журналів одразу;

- зручний для тестування з підтримкою PHPUnit.

Основні недоліки цього фреймворку:

- швидкість розробки Laravel трохи нижча в порівнянні з іншими серверними фреймворками;

- обмежена вбудована підтримка в порівнянні з Django та Ruby on Rails, хоча це долається за допомогою інструментів сторонніх розробників;

- необережні оновлення ризиковані без заходів обережності.

2.3.2 Django

Один з лідуючих на ринку фреймворк із відкритим кодом, заснований на мові програмування Python, відповідає шаблону MVC.

Django підходить для розробки складних і багатофункціональних вебдодатків, керованих базами даних. Забезпечує оптимальне підключення, скорочення розміру кодування, можливість повторного використання та швидку розробку. Використовується Python для всіх операцій цього фреймворку, надає додатковий інтерфейс адміністратора для створення, читання, оновлення та видалення операцій.

Основні переваги цього фреймворку:

- простий для вивчення та використання, що надає можливість розробникам пришвидшити процес освоєння та розробки;

- широкий набір функцій, які допомагають користувачам виконувати деякі загальні вимоги веброботки такі аутенфікація, карти сайту, адміністрування вмісту та ін.;

- безпечна структура, яка допомагає своїм користувачам запобігати проблемам безпеки, включаючи клікджекінг, впровадження SQL і підробку запитів;

- висока масштабованість що відповідає високим вимогам великих проектів;
- універсальний, підходить для розробки широкого спектру застосунків.

Основні недоліки цього фреймворку:

- не підходить для швидких та невеликих проектів, адже вимагає багато часу для програмування, більшої пропускну здатності та часу обробки;
- відсутність умовностей створює проблеми, по типу низького рівню комфорту, контрастні компоненти та повільну швидкість розробки[9].

2.3.3 Express.js

Також відомий як Express, є структурою вебдодатків(каркасом) серверної частини вебзастосунків для Node.js з відкритим кодом, доступний за ліцензією MIT. Використовується для створення API і вебзастосунків і вважається стандартною серверною структурою Node.js.

Express є бекендом для програмного стека MEAN, разом з базою даних MongoDB і каркасом AngularJS для фронтенду[10].

Основні переваги цього фреймворку:

- швидке програмування на стороні сервера;
- більш складний механізм маршрутизації порівняно з Node.js, здатний обробляти динамічні URL-адреси;
- має власний відлагоджувач(дебагер), який дозволяє розробникам легко визначати причини помилок програми.

Основні недоліки цього фреймворку:

- відсутність попередньо визначеного аналізатора тіла;
- не пропонує нічого надзвичайного в плані безпеки;
- не підходить для великих проектів, бо є лише відносно базовим бекенд фреймворком.

2.4 Вибір фреймворків

Після проведеного аналізу популярних фреймворків, а також враховуючи попередній досвід роботи, було вирішено обрати Vue.js як фронтенд і Laravel як бекенд для вебзастосунку.

2.5 Редактор коду Visual Studio Code

Visual Studio Code – це безкоштовний редактор коду представлений Microsoft ще у 2015 році, який надає розробникам велику кількість інструментів, які потрібні для швидкого циклу створення коду-налагодження.

За основу для Visual Studio Code було використано напрацювання вільного проекту Atom, що розвивається компанією GitHub. Зокрема Visual Studio Code є надбудовою над Atom Shell, що використовує браузерний рушій Chromium і Node.js[11].

Редактор містить у собі інструменти роботи з гітом, засоби рефакторингу, навігації по коду, контекстні підказки, вбудований інструмент для тестування і виправлення коду. Дозволяє встановлення розширень які поліпшують роботу і написання коду.

Підтримуються операційні системи Windows, macOS і Linux, більш детальна інформація щодо вимог підтримуваних версій можна знайти на офіційному сайті. Підтримка більшої кількості мов програмування.

Основні переваги цього редактору коду:

- вбудований термінал;
- підтримка Git;
- невеликий розмір;
- безкоштовний;
- велика база користувачів, що полегшує пошук відповіді на будь-яке

запитання яке може виникнути до цього редактору.

Основні недоліки цього редактору коду:

- іноді повільний на системі Linux.

2.6 Вибір системи управління базою даних

MySQL — безкоштовна реляційна система управління базами даних яка є стандартним вибором для невеликих проектів. Розробку та підтримку сайту MySQL здійснює компанія Oracle. MySQL набула в інтернеті широке поширення як система зберігання даних у сайтів, через свою надійність та популярність, переважна більшість сайтів зберігають свої дані в базі MySQL.

У рейтингу Stack Overflow MySQL займає перше місце, що приводить до висновку що програмісти найбільше задають питання, пов'язані саме з MySQL.

У всіх інших рейтингах MySQL впевнено займає другу строчку, і це один із найбільш стабільних результатів серед всіх систем управління базами даних.

Microsoft SQL Server — це система управління базами даних, розроблена компанією Microsoft. Її активно використовують в корпоративному секторі, особливо у великих компаніях. І це не просто СУБД — це цілий комплекс програм, що дозволяє не тільки зберігати, модифікувати й аналізувати дані, а й забезпечувати повну безпеку цих даних і багато іншого.

Oracle Database — це система управління базами даних від компанії Oracle. Ця СУБД також активно використовується великими компаніями і стоїть немалих грошей, але натомість вона надає перевірений роками функціонал і надійність. Тому Oracle Database і Microsoft SQL Server є одними з найсерйознішими конкурентами.

Результати Oracle в рейтингах не такі однозначні, наприклад, за версією PYPL і DB-Engines, Oracle Database займає перше місце, за версією Stack Overflow Oracle знаходиться на восьмій позиції, а за результатами опитувань софтверних компаній Oracle на четвертому місці за популярністю.

PostgreSQL — це безкоштовна реляційна система управління базами даних. Її активно використовують на UNIX-подібних платформах для реалізації як малих та середніх, так і великих проектів.

Найкращий результат у PostgreSQL в рейтингах за версією Stack Overflow, де вона займає другий рядок після MySQL.

PYPL і DB-Engines ставлять PostgreSQL на четверте місце. Опитування показують, що PostgreSQL стоїть на третьому місці за популярністю.

MongoDB — це документоорієнтована система управління базами даних із відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB класифікується як NoSQL і використовує JSON-подібні документи і схему бази даних.

Якщо всі чотири попередніх СУБД на слуху у більшості IT-фахівців, то про MongoDB чули не всі. Але вона впевнено займає 5 сходинку практично у всіх рейтингах[12].

Після збору інформації та аналізу було вирішено використовувати MySQL як систему управління базою даних для створюваного вебзастосунку.

2.7 Діаграма взаємодії клієнта і сервера

Взаємодія клієнту і сервера буде виконуватися у стандартному вигляді запитів POST і GET, вебзастосунком створеному на бекенд фреймворку Laravel.

Процес активності можна розглянути в діаграмі активності (рис. 2.1).

В процесі приймає участь веббраузер, який виконує роль клієнта, вебзастосунок створений за допомогою фреймворків Laravel і Vue.js розташований на сервері і база даних.

Запити клієнта оброблюються фільтрами Laravel, які знаходяться у `/app/http/middleware`. Для створення власного фільтра використовується команда `php artisan make middleware {ім'я створюваного фільтра}`.

Контролери знаходяться в `app/http/controllers/{ім'я контролера}`.

Команда для швидкого створення шаблону авторизації `php artisan make:auth`. Файл конфігурації знаходиться в `/config/auth.php`.

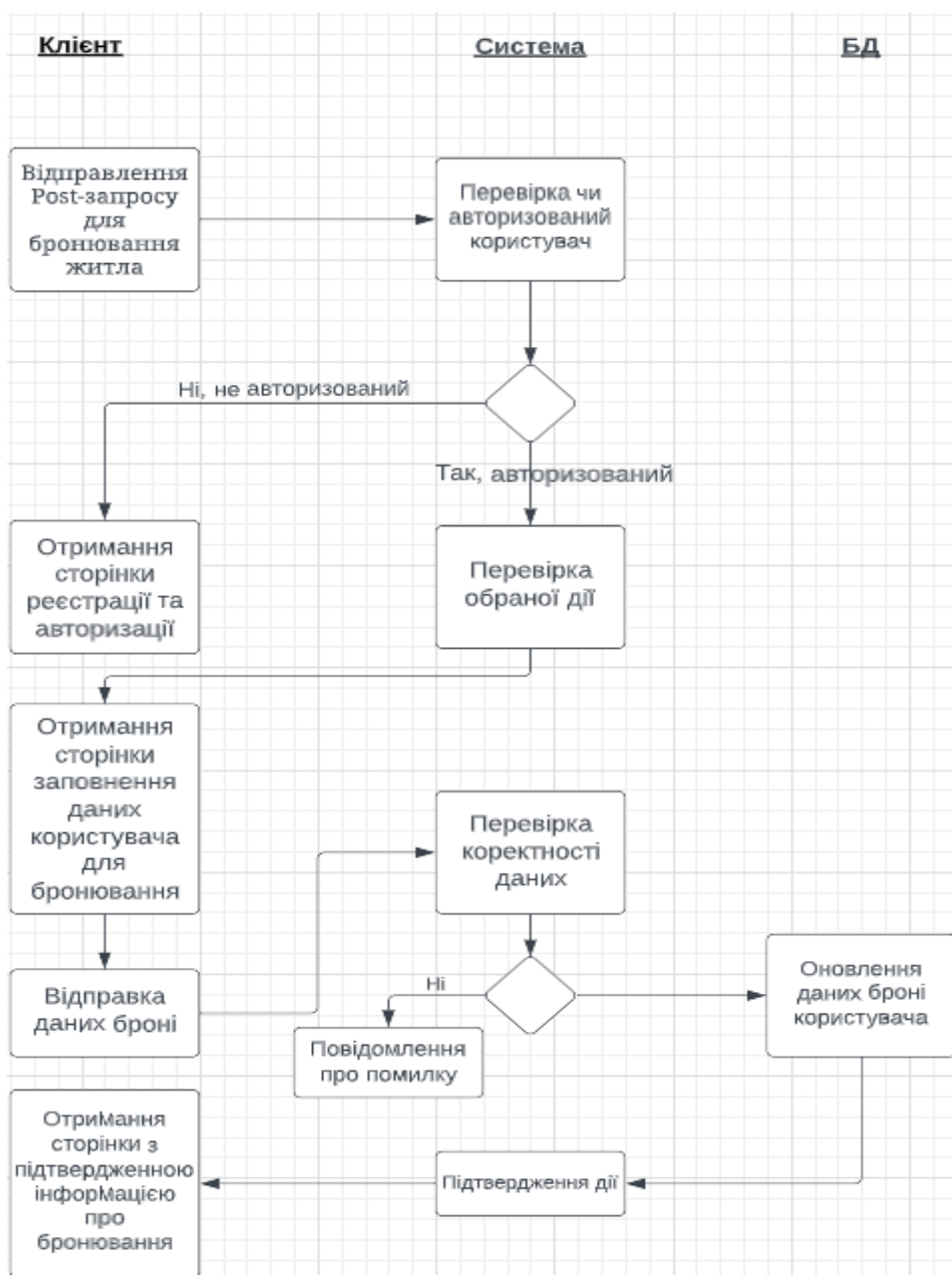


Рисунок 2.1 – Діаграма активності взаємодії клієнта і сервера

Після відправлення запиту на бронювання, фільтр обробляє запит серверу, і відправляє користувача або на сторінку реєстрації та авторизації в разі якщо він не авторизований, або запит пройде далі. Після того, як на сервері пройде перевірка обраної дії, користувач отримує сторінку для заповнення даних для бронювання й після її заповнення йде запит на сервер з перевіркою коректності даних, в разі некоректно введених даних буде показана помилка, при правильному наборі даних, в базу даних пропускається інформація, після

підтвердження дії користувач побачить сторінку з підтвердженням інформації про бронювання.

2.8 Діаграма дій користувача у системі

Активність прецедента при «Оформленні бронювання» (рис. 2.2).

Основні дії:

- користувач переходить на сторінку авторизації;
- користувач вводить свої дані в поля логіну та пароллю, натискає «авторизація». Йде обробка запиту авторизації, перевірка чи усі поля заповнені та правильність інформації. При перевірці якщо не всі поля заповнені то альтернативний потік A1, якщо всі, то йде перевірка наявності такого користувача в системі, якщо не – альтернативний потік A2, якщо є, то авторизація проходить успішно й переходить до бронювання житла;
- користувач переходить на сторінку оформлення бронювання;
- користувач вводить дані необхідні для бронювання, йде перевірка заповнення усіх полів, якщо не всі заповнені, альтернативний потік A3, якщо всі то йде перевірка правильності даних, якщо неправильно то альтернативний потік A4, якщо все правильно система підтверджує інформацію бронювання і користувач у своєму профелі буде бачити статус свого бронювання.

Альтернативний потік.

A1. Користувач побачить системне повідомлення про помилку «Помилка. Не всі поля заповнені.». Відбувається перехід до кроку 1 основного потоку.

A2. Користувач побачить системне повідомлення «Помилка. Користувача з такими даними не знайдено. Перевірте правильність даних.». Відбувається перехід до кроку 1 основного потоку.

A3. Користувач бачить системне повідомлення «Помилка. Не всі поля заповнені.». Відбувається перехід до кроку 4 основного потоку.

А4. Користувач бачить системне повідомлення «Помилка. {поле} Заповнено не коректно.». Відбувається перехід до кроку 4 основного потоку.

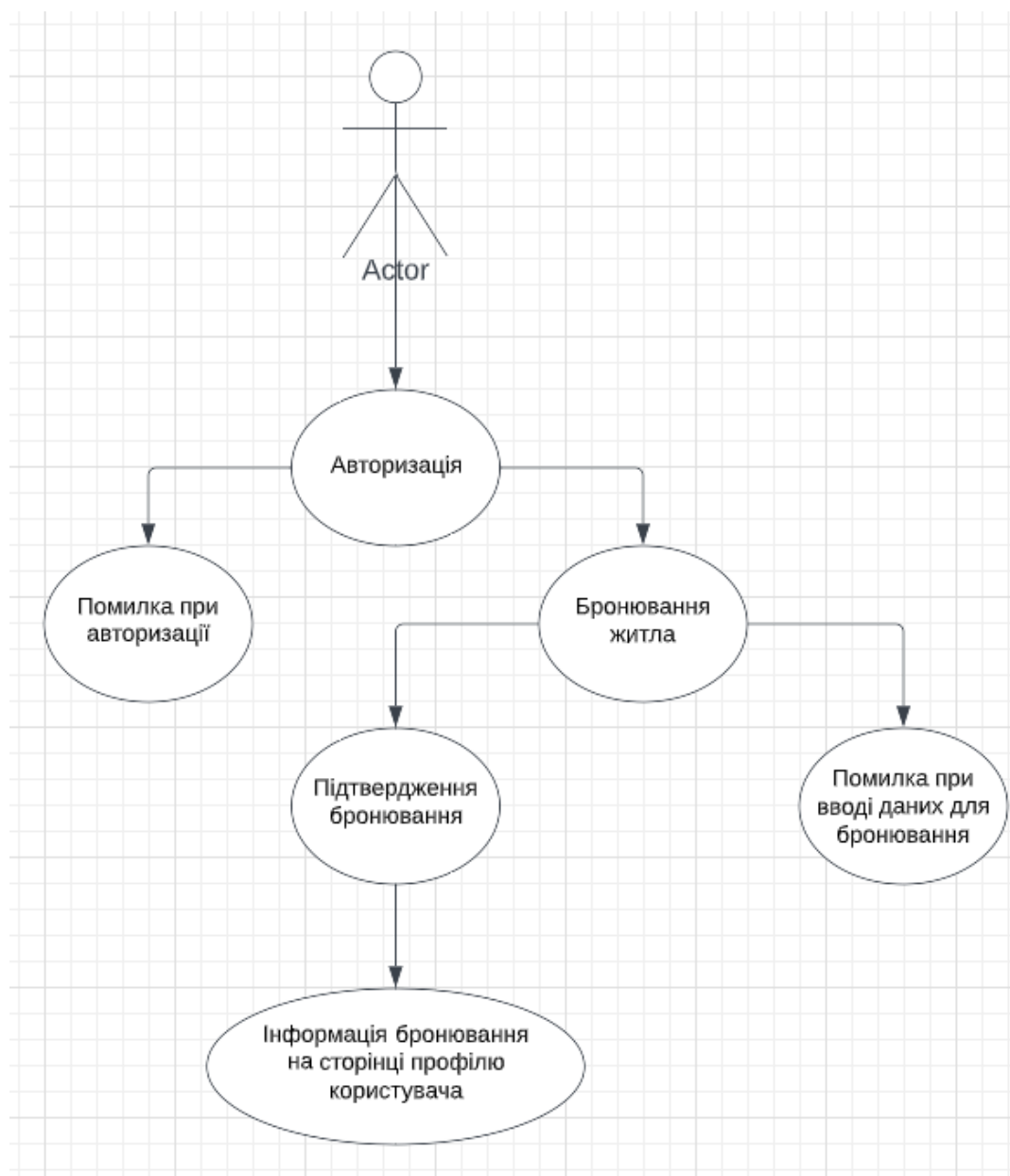


Рисунок 2.2 – Діаграма прецедента «Оформлення броні»

2.9 Розробка дизайну сайту

Основою для дизайну сайту було обрано створити вайрфрейм, чорно-білий макет якій буде вміщувати в собі структуру та наповнення проекту (рис. 2.3).

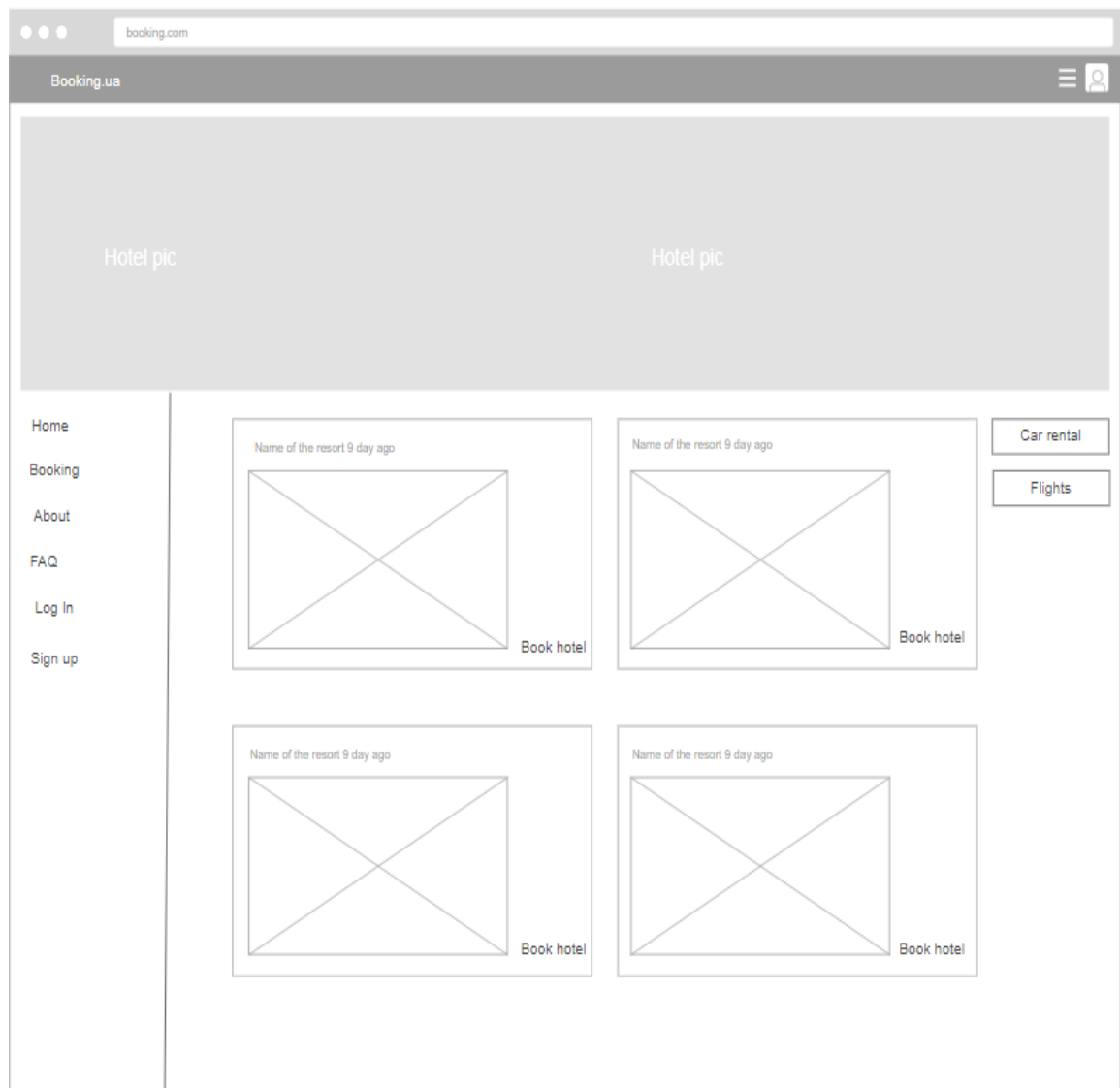


Рисунок 2.3 – Вайрфрейм головної сторінки

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Опис структури файлів проєкту

Структуру зображено на (рис. 3.1).

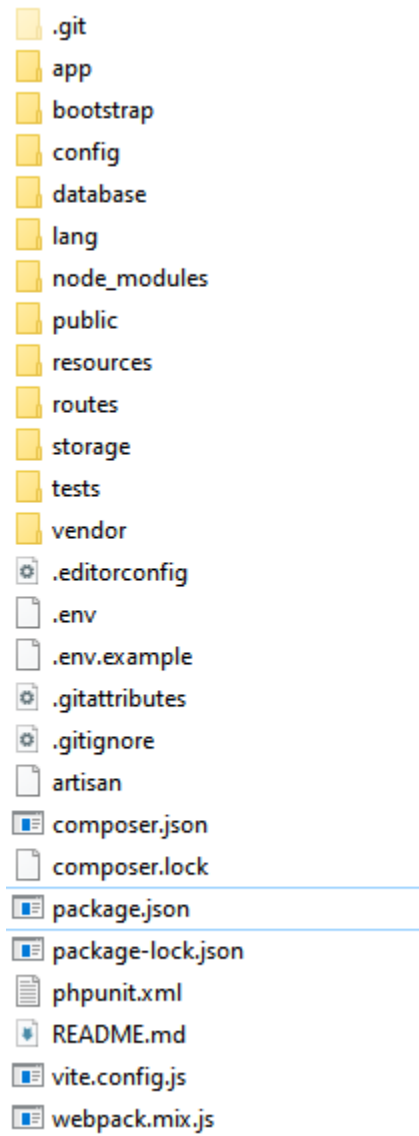


Рисунок 3.1 –Структура проєкту

Структура проекту складається з:

- `.env` – файл який містить інформацію для роботи та підключення бази даних;
- `Artisan` – утиліта для виконання різних службових задач сервера;
- `App` – папка, яка містить у собі контролери `app/Controllers` та фільтри `app/Middleware`;
- `App.blade` – файл який містить у собі бекенд застосунку;
- `Bootstrap` – каталог який містить у собі файл `app.php`, який запускає фреймворк. В каталозі також знаходиться каталог кеша, який містить сгенеровані файли для оптимізації кеш маршрутів й служб;
- `Config` – папка яка містить конфігурацію вебдодатку;
- `Database` – папка яка містить у собі базу даних з файлами міграції;
- `Httpaccess` – файл конфігурації вебсервера `Apache`, який надає контроль над роботою сервера та настройками вебзастосунку;
- `Web.config` – файл який визначає поведінку додатків `ASP.NET`. `Microsoft .NET.Framework`;
- `Resources` – містить в собі шаблони, а також необработані, незакомпільовані ресурси, наприклад `JavaScript` або `CSS`.
- `Routes` – містить всі визначення маршрутів для застосунку. ПО замовчуванню `Laravel` має в собі декілька файлів маршрутів, а саме `web.php`, `api.php`, `console.php` і `channels.php`;
- `Storage` – каталог який містить журнали(логи), скомпільовані шаблони `Blade`, файли сесій, кеш тощо.
- `Tests` – файл який містить автоматизовані тести. Приклади модульного `PHPUnit` і функціональні тести надаються при установці `Laravel`.
- `Vendor` – містить `Composer`-залежності.

3.2 База даних

Даний вебзастосунок використовує програму phpAdminer для роботи з базою даних. Laravel підтримує чотири бази даних, MySQL 5.7+, PostgreSQL 9.6+, SQLite 3.8.8+ та SQL Server 2017+. Повне налаштування сліжб баз даних знаходиться в файлі `config/database.php`, в цьому файлі можна визначити всі з'єднання до бази даних, а також вказати, яке з'єднання буде використовуватися за замовчуванням.

За контроль версій бази даних буде відповідати конструктор таблиць `migrations` разом з міграціями. Це надає повний контроль над змінами в структурі бази даних й дозволяє іншим учасникам бачити зміни. Конструктор сам по собі є незалежним від бази даних інтерфейсом, який маніпулює таблицями.

Міграції розташовані в стандартному місці після встановлення Laravel, в папці `database/migration`. Для генерації міграції використовується команда `make:migration` Artisan. Ця команда помістить новий клас міграції в каталог вебзастосунку. Кожне ім'я файлу міграції має тимчасову мітку, яка дозволяє Laravel визначати порядок застосування міграцій.

Для запуску усіх незавершених міграцій виконується команда `php artisan migrate`. Для відкату останньої операції міграції можна виконати команду `php artisan migrate:rollback`, що відкатує останній пакет міграцій, котрий може складатися з декількох файлів міграції[13].

У файлі `.env` знаходяться налаштування бази даних:

`DB_HOST=localhost` або `127.0.0.1`

`DB_DATABASE=ім'я бази даних`

`DB_USERNAME=логін бази даних`

`DB_PASSWORD=пароль бази даних`

Далі робимо міграцію за допомогою команди: `php artisan make:migration users` для створення таблички бази даних яка буде вміщувати інформацію

користувача (рис. 3.2), після чого у папці /app/Models/ було створено модель users.php, а у database/migrations users_table.php. Таким же чином робимо міграції apartments, apartment_services, services, photo, rents.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
use App\Models\User;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->enum('role', User::ROLES)->default(User::ROLE_CUSTOMER);
            $table->string('email')->unique();
            $table->string('avatar')->nullable();
            $table
                ->enum('country_code', array_keys(User::COUNTRY_CODES))
                ->default(User::DEFAULT_COUNTRY_CODE);
            $table->string('phone1')->nullable();
            $table->string('phone2')->nullable();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }
};
```

Рисунок 3.2 – Код таблиці з інформацією користувача [9]

База даних вебзастосунку складається з основних таблиць, перелік яких наведено в (табл. 3.1-3.6).

Таблиця 3.1 – Структура таблиці users

Назва поля	Призначення	Тип даних
id	Номер в базі даних	bigint unsigned
name	Ім'я користувача	string
email	Пошта	string
phone1	Перший номер мобільного	string
phone2t	Другий номер мобільного	string
timestamp	Дата реєстрації	timestamp
password	Пароль	string

Таблиця 3.2 – Структура таблиці apartments

Назва поля	Призначення	Тип даних
id	Номер в базі даних	bigint unsigned
name	Назва готелю або хостелу чи апартаментів	string
type	Тип апартаментів	enum
short_description	Короткий опис апартаментів	string
description	Повний опис апартаментів	string
price	ціна	integer
address	адреса	string
available	доступність	boolean
multiple	Кількість мість	boolean
tenants	орендарі	smallInteger

Таблиця 3.3 – Структура таблиці apartment_services

Назва поля	Призначення	Тип даних
id	Номер в базі даних	bigint unsigned
apartment_id	Номер апартаментів	bigint unsigned
services_id	Номер сервісів	bigint unsigned

Таблиця 3.4 – Структура таблиці photo

Назва поля	Призначення	Тип даних
id	Номер в базі даних	bigint unsigned
apartment_id	Номер апартаментів	bigint unsigned
name	Назва фото	string
desription	опис	string
path	шлях до фото	string
main	Флаг для головного фото	boolean

Таблиця 3.5 – Структура таблиці services

Назва поля	Призначення	Тип даних
id	Номер в базі даних	bigint unsigned
name	Назва послуги	string
description	Опис послуги	string
icon	Іконка послуги	string

Таблиця 3.6 – Структура таблиці rents

Назва поля	Призначення	Тип даних
id	Номер в базі даних	bigint unsigned
user_id	Ід користувача який орендує ці апартаменти	bigint unsigned
apartment_id	Номер апартаментів в системі	bigint unsigned
arrival_date	Дата приїзду орендатора	timestamp
departure_date	Дата від'їзду орендатора	timestamp
paid	Чи проплачено наперед	boolean
price	ціна	integer
review_grade	Рейтинг відгуку	integer
review	відгук	string
owner_review_grade	Рейтинг відгук власника апартаментів	integer
owner_review	Відгук власника апартаментів	string

Для зберігання даних використовується MySQL.

3.3 Логіка обробки HTTP-запитів та її шаблони

Всі шаблони відображення знаходяться на клієнтській частині(фронтенд), тобто відображення та рендеринг даних відбуваються без задіювання серверної частини, при переході по силці клієнтська сторона робить запит на бекенд для отримання лише тих даних в форматі json, які потрібні тільки для відображення в даний момент. Це являється основною перевагою SPA, тому що дані запрошуються лише ті що необхідні, що призводить до

пониження навантаження на сервер, та зменшення інтернет трафіку, та також відбувається фактична перезагрузка сторінки а лише тієї частини, яка відповідає поточній адресі URL. Таким чином вебсервер виконує такі функції:

- збереження даних;
- контроль доступу користувача до своїх даних;
- та функції CRUD;
- валідація даних.

Використовується шаблонізатор Blade, який входить до складу Laravel.

Основні шаблони проекту:

- `app.blade.php` – основний шаблон що повертається з бекенду, після загрузки шаблону весь контроль переходить до фреймворку Vue.js (фронтенд);
- `Apartment.vue` – шаблон, що відповідає за показ одного конкретного об'єкту оренди, коли користувач натискає кнопку зарезервувати;
- `BookingList.vue` – шаблон, відповідаючий за показ списку зарезервованих помешкань, відміну старих резервувань та тих, що проходять процес підтвердження;
- `Faq.vue` – шаблон сторінки з відповідями на часті запитання користувачів по типу способів оплати, типів номерів чи як скасувати бронь;
- `Settings.vue` – шаблон відповідаючий за налаштування профілю користувача та його інформацією як ім'я, пошта, номери телефонів тощо;
- `ApartmentsList.vue` – шаблон головної сторінки, на якій відображено перелік помешкань для оренди.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Призначення та умови застосування вебзастосунку

Вебзастосунок для оренди житла виконаний у вигляді SPA додатку на фреймворках Laravel та Vue.js.

Цей вебдодаток надає можливість користувачам робити бронювання житла вибираючи серед основних варіантів на головній сторінці та бачити статус свого бронювання прямо на вкладці мої бронювання, є можливість налаштувань, подивитися пам'ятки, та інформація контактів.

Для використання сервісу, його необхідно запустити через віртуальний сервер, або завантажити на хостинг, який має відповідати певним вимогам:

- .PHP ≥ 5.4 ;
- Mcrypt PHP Extension;
- OpenSSL PHP Extension;
- Mbstring PHP Extension.

Мінімальні системні вимоги до заліза сервера було обрано серед сучасних систем різних провайдерів віртуальних приватних серверів:

- оперативна пам'ять 8 GB;
- 4vCPUs;
- 50 GB пам'яті жорсткого диску SSD.

4.2 Налаштування та підключення за допомогою OpenServer

OpenServer – це спеціальних сервер призначений для тестування та розробки вебпроектів. Він включає в себе такий комплекс програм:

- Apache – надійний та гнучкий сервер з відкритим кодом;
- PHP – популярна мова програмування призначена для веброботи;
- PHPMyAdmin – спеціальний інструмент побудований на мові PHP;
- Nginx – ще один гнучкий сервер;
- MySQL – система, призначена якої керування базами даних.

Для початку роботи із сервером, необхідно його скачати з офіційного сайту. Закачати можна безкоштовно але тоді швидкість буде повільна закачування.

Після скачування та установки потрібно буде перейти в місце установки серверу, де буде необхідна папка domains, в якій буде зберігатися проєкт, та сам OpenServer.exe файл завдяки якому буде запускатися сервер (рис. 4.1.).

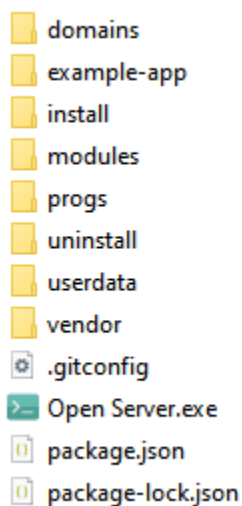


Рисунок 4.1 – Структура папки з сервером

Після запуску серверу в області повідомлень(трей) ком'ютера з'явиться червоний прапорець який є панеллю управління OpenServer в якій відбувається

керування сервером, після налаштувань та запуску зміниться на зелений прапорець (рис. 4.2).

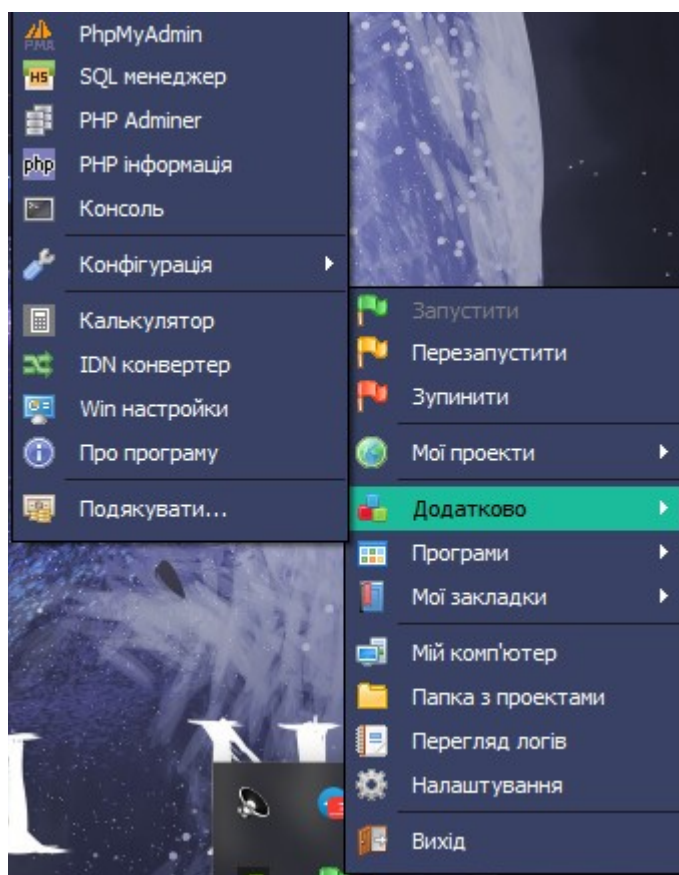


Рисунок 4.2 – Панель управління сервером

Для створення нового проекту достатньо перейти до папки domains і створити там папку якій задається потрібна назва. Для перевірки роботи серверу достатньо створити документ index.php, та в ньому ввести заголовок h1. Після цього перезапускаємо сервер та перевіряємо наявність нового сайту або вебзастосунку перейшовши в меню мої проекти.

В застосунку OpenServer є велика кількість налаштувань, як версії Apache, PHP, MySQL тощо. Налаштування портів, ір-адреси, обирання корневої папки доменів. Які необхідні для повної підтримки застосунку (рис. 4.3).

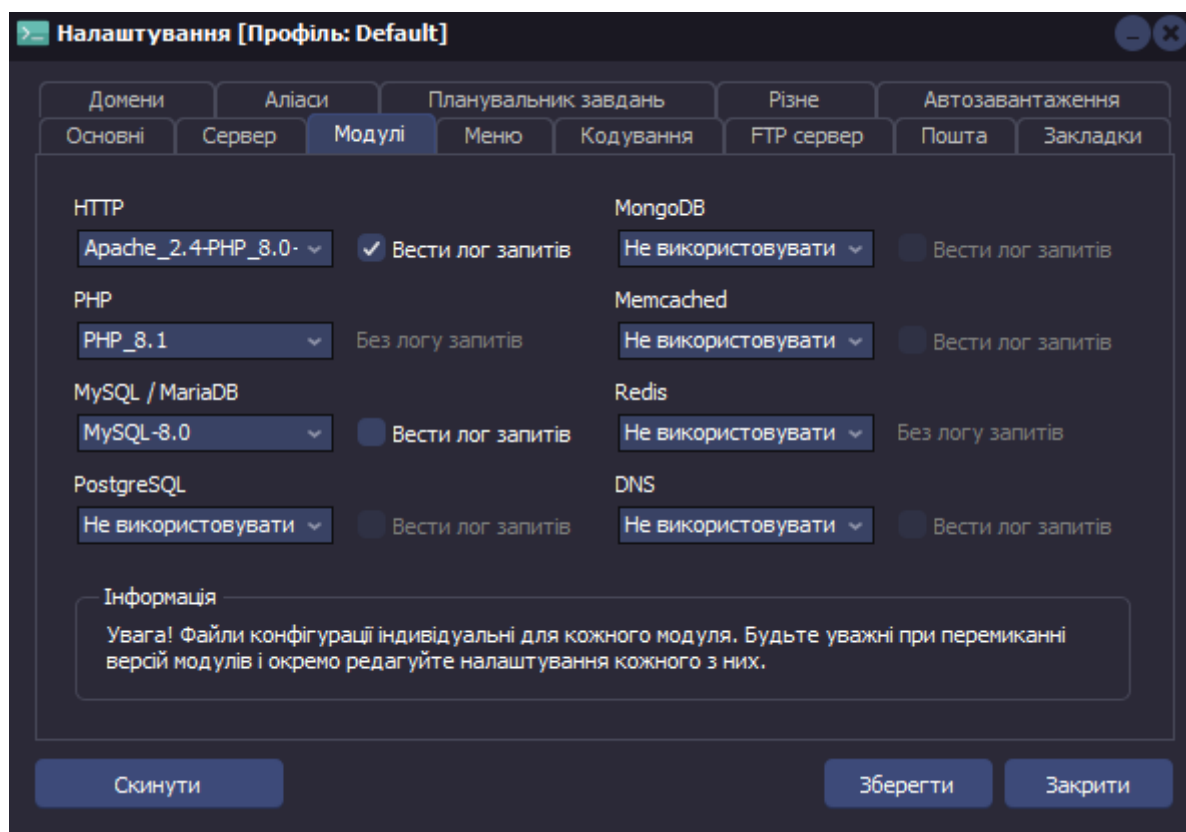


Рисунок 4.3 – Панель налаштувань сервера

Основними особливостями OpenServer можна відзначити:

- можливість роботи з флеш-накопичувача;
- домен створюється завдяки створенню простої папки;
- захист сервера від зовнішнього втручання;
- швидкі доступи до модулів та доменів;
- можливість роботи з кириличними доменами.

Інформація бази даних та дані необхідні для входу в неї прописуються в файлі `.env`. Саму базу даних та її таблиці можна продивитися за допомогою PHP Adminer, який вбудований до OpenServer, вхід до управління якого здійснюється завдяки даним з файлу `.env` (рис. 4.4 – рис. 4.7).

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=hotel
DB_USERNAME=root
DB_PASSWORD=root
```

Рисунок 4.4 – Налаштування з файлу .env

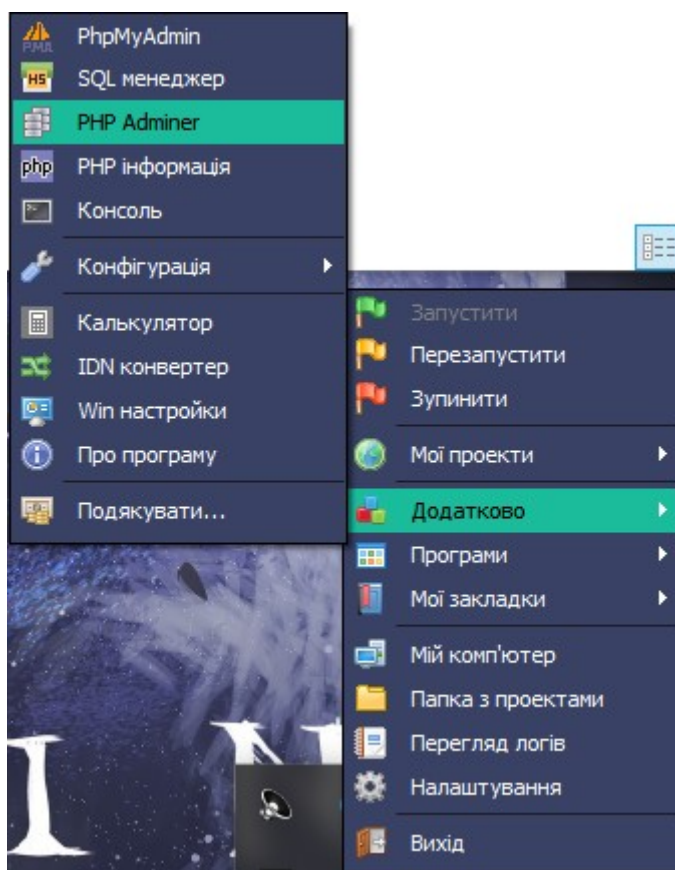


Рисунок 4.5 – PHP Adminer в меню OpenServer

Мова: Українська ▼

Adminer 4.7.8 4.8.1

Увійти

Система Бази Даних	MySQL ▼
Сервер	localhost
Користувач	root
Пароль	
База даних	

☐ Пам'ятати сесію

Рисунок 4.6 – Вхід до PHP Adminer

Мова: Українська ▼ MySQL » Сервер

Adminer 4.7.8 4.8.1

Обрати базу даних

DB: ▼

SQL запит
 Імпортувати
 Експорт

Створити базу даних Привілеї Перелік процесів
 Змінні Статус

Версія MySQL: **8.0.19** з PHP-розширенням **MySQLi**
 Ви увійшли як: **root@localhost**

	База даних - Оновити	Співставлення	Таблиці	Size - Compute
☐	diplom	utf8_general_ci	?	?
☐	hotel	utf8_general_ci	?	?
☐	information_schema	utf8_general_ci	?	?
☐	mysql	utf8mb4_0900_ai_ci	?	?
☐	performance_schema	utf8mb4_0900_ai_ci	?	?
☐	sys	utf8mb4_0900_ai_ci	?	?

Selected (0)

Рисунок 4.7 – Меню управління PHP Adminer

Після перевірки працездатності проекту та бази даних переходимо до перевірки працездатності вебзастосунку через OpenServer (рис. 4.8).

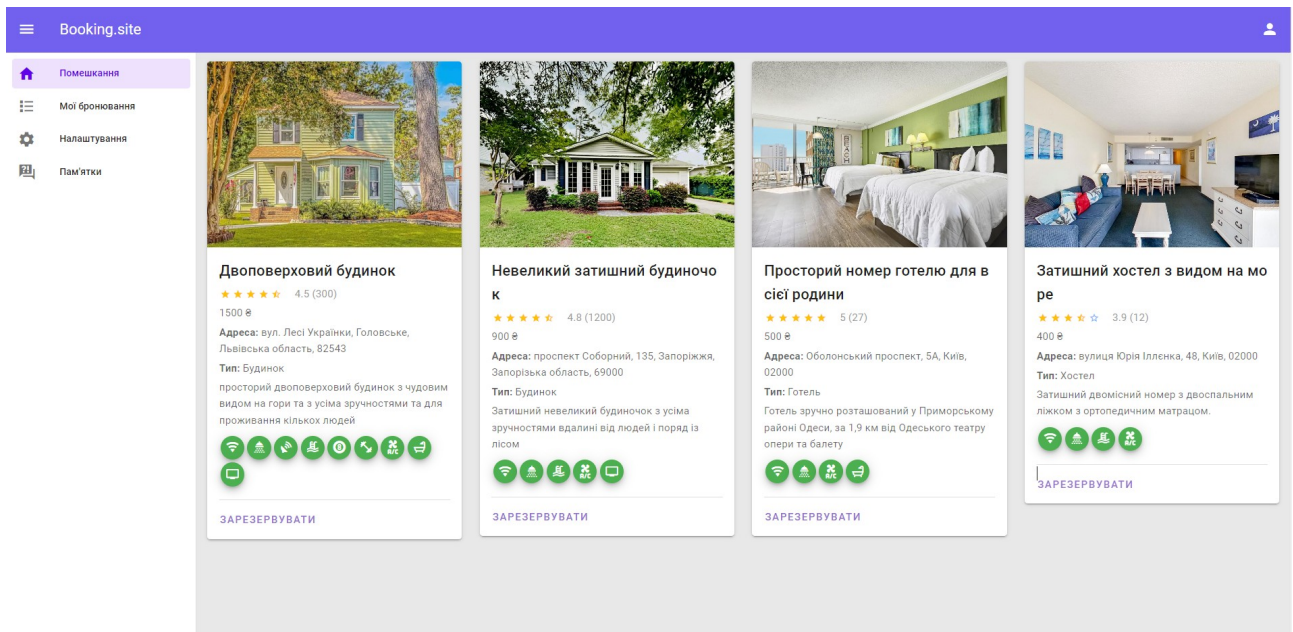


Рисунок 4.8 – Результат запуску вебзастосунку через OpenServer

4.3 Робота з вебзастосунком

Для роботи з вебзастосунком достатньо мати доступ до пристрою який підтримує інтернет зв'язок та відповідає мінімальним вимогам для веббраузера.

Для запуску вебзастосунку включаємо OpenServer, та запускаємо вебзастосунок, пропускаючи в терміналі `php artisan serve`. Видає айпі адресу яку вводимо в адресну строку браузера, після чого потрапляємо на логіну (рис. 4.9).

На сторінці авторизації вже зареєстрований користувач просто вводить свої дані та заходить до застосунку, а не зареєстрований може зареєструватися нажавши на кнопку реєстрація.

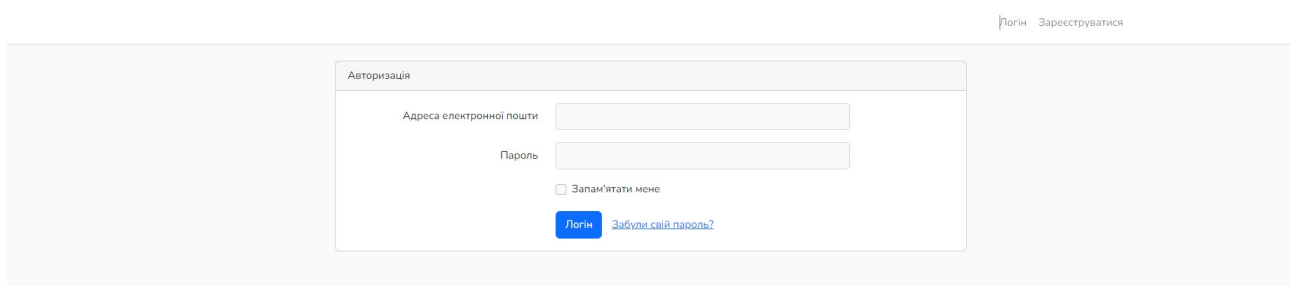


Рисунок 4.9 – Сторінка авторизації

На головній сторінці сторінці користувач в меню зліва має доступ до вкладок з різними варіантами для оренди помешкання, його бронювання, налаштування свого профілю та пам'ятка з частими запитаннями (рис. 4.10).

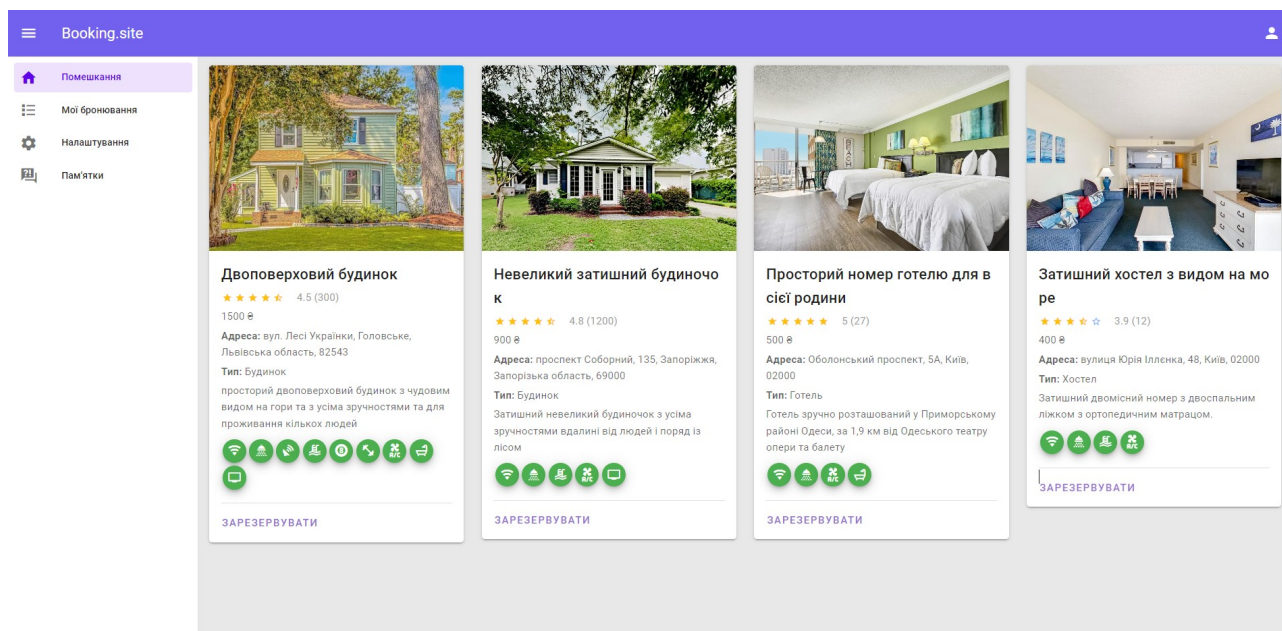


Рисунок 4.10 – Головна сторінка вебзастосунку

Щоб орендувати помешкання яке сподобалось, необхідно під необхідною для користувача пропозицією натиснути на кнопку зарезервувати, тоді відкриється повний опис цієї пропозиції, з відображенням ціни за добу, ціни за обрану дату проживання, сервісами які надаються в даній пропозиції (рис.4.11).

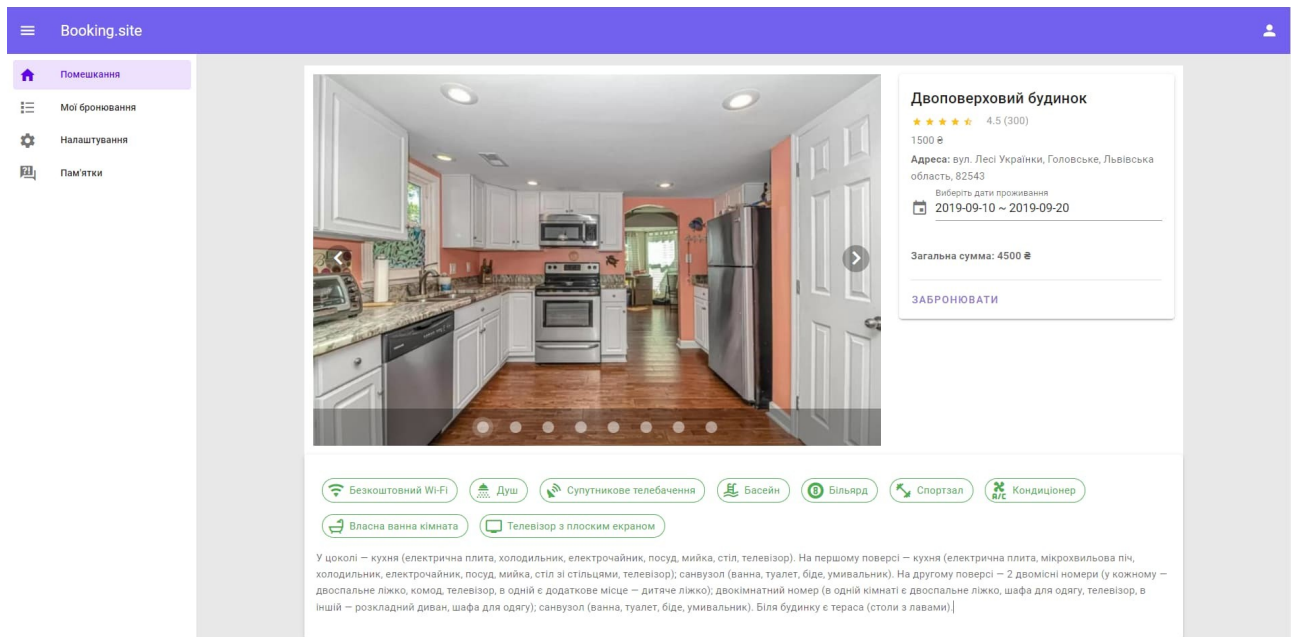


Рисунок 4.11 – Сторінка резервування

Після резервування користувач переходить до вкладки мої бронювання де він побачить статуси своїх бронювань, та буде можливість скасувати бронювання які знаходяться у статусі виконується або в процесі підтвердження (рис.4.12).

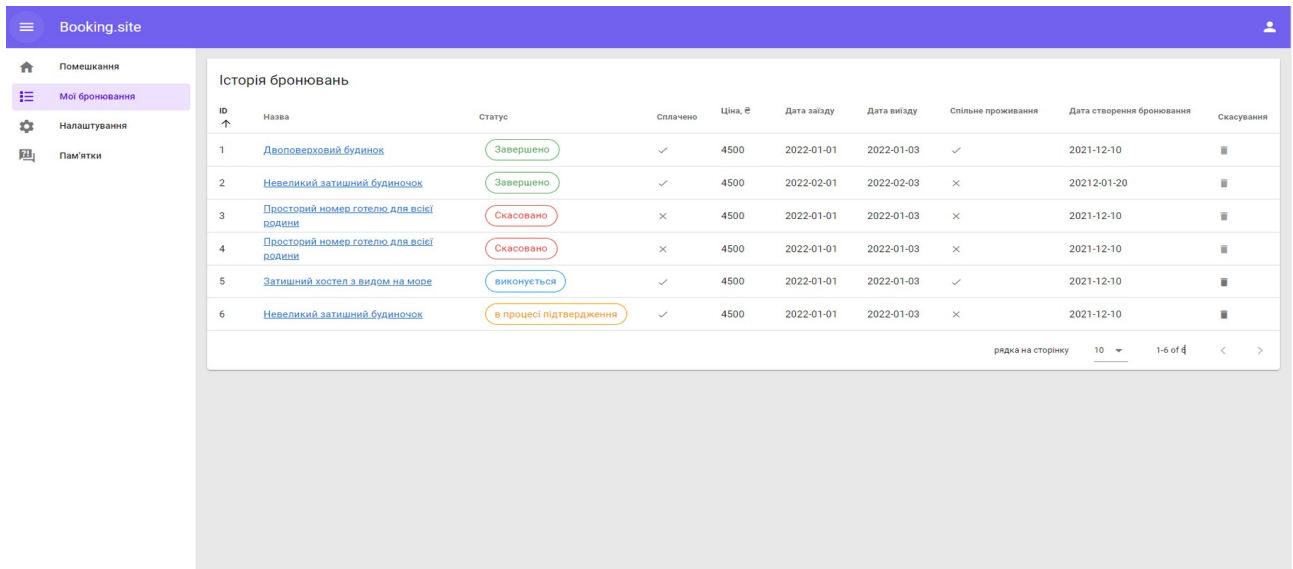


Рисунок 4.12 – Сторінка Мої бронювання

На цій же сторінці можна додати власний відгук (рис. 4.13).

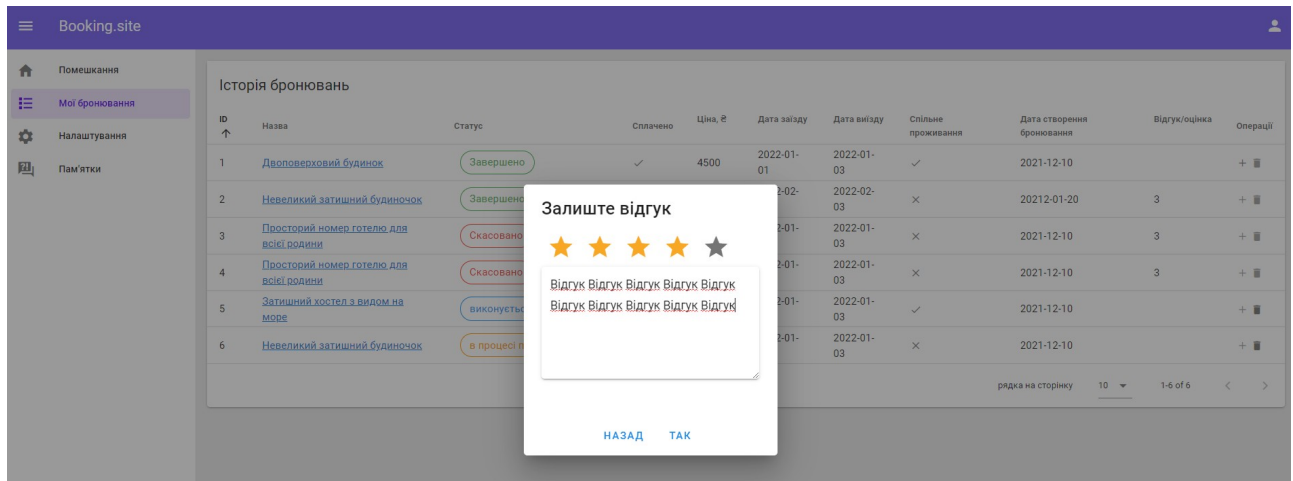


Рисунок 4.13 – Сторінка Мої бронювання, додавання відгуку

Ще доступна сторінка налаштування, на якій користувач має доступ до своєї інформації, яку він може змінювати в будь який час (рис. 4.14).

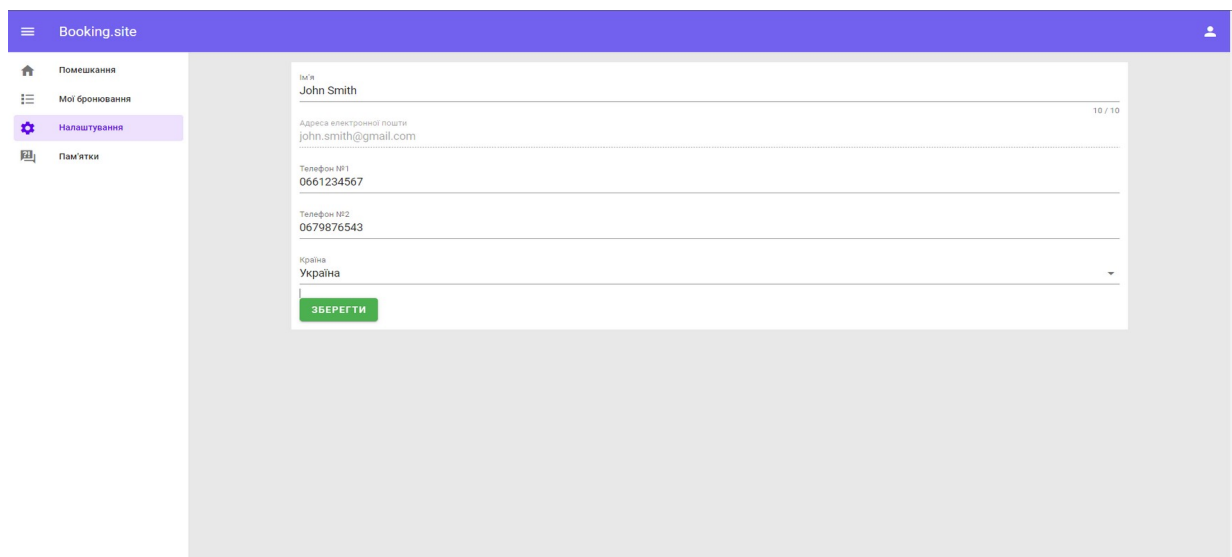


Рисунок 4.14 – Сторінка Налаштування

Остання та не менш важлива сторінка пам'ятки, на якій користувач може знайти відповіді на найчастіші запитання до вебзастосунку (рис. 4.15).

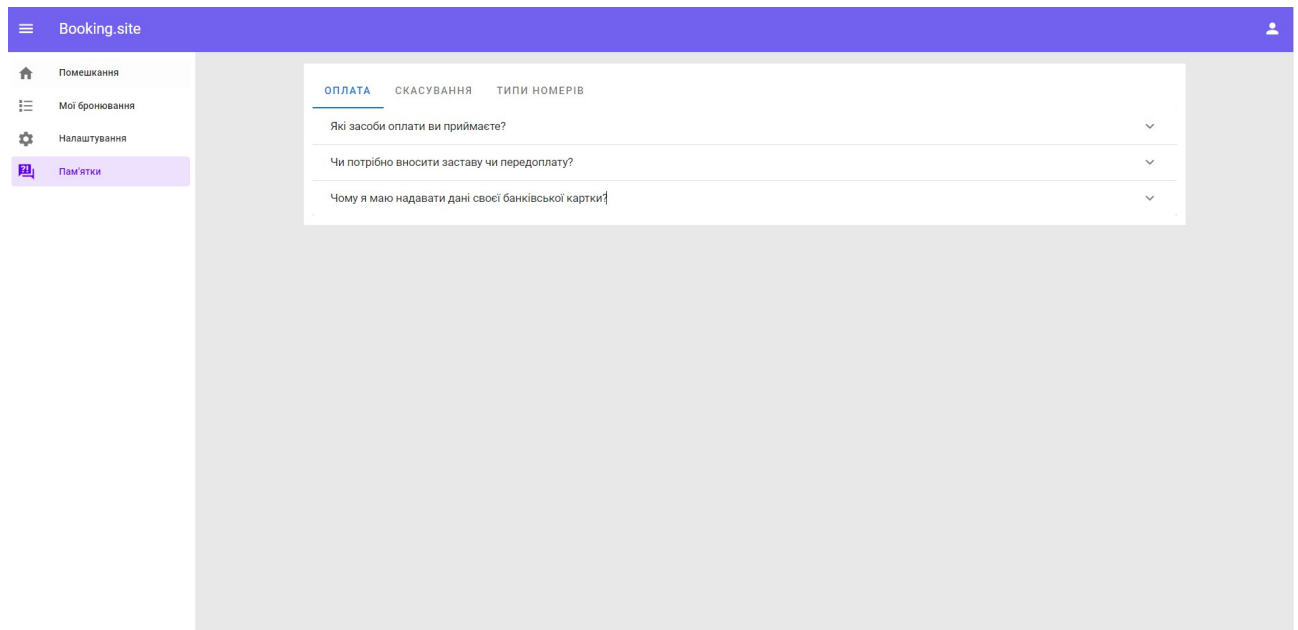


Рисунок 4.15 – Сторінка Пам'ятки

4.4 Можливі модифікації

Для покращення роботи з вебзастосунком, необхідно:

- Геолокація користувача для пошуку найближчих варіантів оренди;
- Служба підтримки онлайн у вигляді чату.

4.5 Вхідні та вихідні дані

Вхідними даними є HTTPS запити, які користувачі відправляються через браузер на сервер вебзастосунку для отримання сторінки.

Вихідними даними є згенеровані HTML сторінки.

ВИСНОВКИ

В ході проектування вебзастосунку для оренди житла було двало вирішено початкові задачі, необхідні для подальшої розробки. Було проведено аналіз предметної області та програмних засобів для створення вебдодатку який буде повністю відповідати вимогам як користувачів так і власнику вебдодатку.

Після аналізу необхідних технологій, було визначено фреймворк Laravel буде відповідати за бекенд, фреймворк Vue.js буде відповідати за фронтенд, середою розробки буде Visual Studio Code.

Далі було розроблено вайрфрейм майбутнього вебзастосунку який надає основне уявлення інтерфейсу сайту, та проведений аналіз дій користувача з цим інтерфейсом.

Далі проведено написання коду програми й тестування продукту й закінчення кінцевої версії.

Як результат було створено вебзастосунок оренди житла за допомогою якого користувачі можуть:

- орендувати готелю,хостелу, частного будинку;
- можуть відслідковуватти стан свого бронювання готелю;
- інформація про номери таксі в місті.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Booking.com [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Booking.com>.
2. Топ сервісів бронювання [Електрон. ресурс]. – Режим доступу: <https://otdatvest.biz.ua/top-10-krashhih-servisiv-bronjuvannja-goteliv/>
3. HTML hypertext Markup Language [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/html>.
4. CSS таблиця каскадних стилів [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/CSS>.
5. PHP [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/PHP>.
6. JQuery [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/JQuery>.
7. Плюси і мінуси фреймворку JQuery [Electronic resource]. – Access mode: <https://monocuped.com/blog/best-front-end-frameworks/>.
8. Back-end framework Laravel [Electronic resource]. – Access mode: <https://blog.back4app.com/backend-frameworks/#Laravel>.
9. Back-end framework Django [Electronic resource] – Access mode: <https://blog.back4app.com/backend-frameworks/#Django>.
10. Express.js [Електрон. ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Express.js>.
11. Visual Studio Code [Електрон. ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Visual_Studio_Code.
12. Топ 5 кращих систем управління базами даних [Електрон. ресурс] – Режим доступу: <https://www.management.com.ua/partners/2021/02/21/5-najpopulyarnishih-sistem-upravlinnya-bazami-danih-2020-roku/>.
13. Laravel міграції [Електрон. ресурс] – Режим доступу: <https://laravel.su/docs/8.x/migrations/>.

ДОДАТОК А
Технічне завдання

Вступ

Вебзастосунок для орендування житла метою якого є надання можливості користувачам резервувати та орендувати житло та бачити стан їх броні.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація застосунку спільної оренди житла», затверджене наказом Національного університету «Запорізька політехніка» № 363 від 4 листопада 2022 р.

A.2 Призначення розробки

Програмний продукт для орендування житла з наданням можливості користувачам резервувати та орендувати житло та бачити стан їх броні за допомогою зручного односторінкового вебзастосунку.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для орендування житла та резервації повинно забезпечити виконання таких функцій:

- реєстрація користувача (для незареєстрованих користувачів);
- авторизація користувача;
- переглядання усіх доступних пропозицій для бронювання;
- резервування житла;
- відслідковування стану своєї броні;
- відміна бронювання;

– зміна інформації в своєму профілі.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс вебзастосунку для резервації та оренди житла має бути простим та зручним у використанні.

A.3.3 Вимоги до надійності

Програмне забезпечення для резервації та оренди житла повинно оброблювати несподівані ситуації до яких можуть призвести дії користувача, пропускна можливість серверної частини програмного забезпечення повинна витримувати певне навантаження без змін у продуктивності.

A.3.4 Умови експлуатації

Програмне забезпечення для резервації та оренди потребує лише підтримки сучасного браузера на пристрої користувача.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації вебзастосунку для резервування та оренди житла необхідно таке апаратне забезпечення:

- серверний комп'ютер із процесором з частотою 1ГГц і вище;
- оперативна пам'ять – не менше 1ГБ;
- вільне місце на жорсткому диску – 1ГБ;
- з сторони клієнта мобільний телефон, ноутбук, планшет з можливістю виходу в інтернет та сучасним браузером.

A.3.6 Вимоги до маркування і пакування

Вебзастосунок для резервації та оренди житла може бути записаний на флешносії або іншому носії інформації. На пакуванні потрібно написати назву програми – «Вебзастосунок для резервування та оренди житла».

A.4 Вхідні дані до роботи

Вхідними даними вебзастосунку для резервації та оренди житла є інформація про користувача (логін, пароль), а також резервація житла, яку він виконує на початку роботи з застосунком.

Вихідними даними програмного забезпечення для організації та підтримки онлайн-чату є дані про його резервацію, що користувач отримує після завершення бронювання.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Вебзастосунок для резервації та оренди житла розроблено за допомогою мови JavaScript, PHP, та фреймворків Vue.js та Laravel.

Б.2 Функціональне призначення

Вебзастосунок для резервації та оренди житла призначено для надання можливостей користувачам резервувати житло, та бачити інформацію по стану своєї броні в реальному часі.

Вебзастосунок для резервації та оренди житла забезпечує виконання таких функцій:

- реєстрація користувача (для незареєстрованих користувачів);
- авторизація користувача;
- просмотр та зміна даних профілю;
- резервації житла;
- переглядання усіх доступних пропозицій для бронювання;
- відслідковування стану своєї броні;
- відміна бронювання;

Б.3 Використані технічні засоби

Для експлуатації вебзастосунку для резервування та оренди житла необхідно таке апаратне забезпечення:

- серверний комп'ютер із процесором з частотою 1ГГц і вище;
- оперативна пам'ять – не менше 1ГБ;
- вільне місце на жорсткому диску – 1ГБ;
- з сторони клієнта мобільний телефон, ноутбук, планшет з можливістю виходу в інтернет та сучасним браузером.

Б.4 Виклик і завантаження

Для розгортання проєкту вебзастосунку для резервації та оренди житла потрібно завантажити на кінцевий вебсервер власне файли проєкту, та прочитати інструкцію за встановлення оточення Vue.js та Laravel (інформація в документаціях на офіційних сайтах <https://laravel.com/docs/9.x> та <https://vuetifyjs.com/en/introduction/why-vuetify/#feature-guides>).

Для зберігання даних вебзастосунку для резервації та оренди житла використовується база даних MySQL. Для її інсталяції потрібно виконати інструкцію на сайті <https://docs.oracle.com/en-us/iaas/mysql-database/doc/getting-started-mysql-database-service.html#GUID-3C1DA346-CDA2-442E-9D22-A7C244B9DF04>.

Після інсталяції програмних продуктів Vue.js, Laravel та MySQL потрібно в командному рядку перейти до директорії проєкту та виконати команду `composer install` та `npm install` (для завантаження додаткових програмних модулів). Після цього виконати команду `php artisan migrate` для завантаження міграцій вебсерверу та команду `php artisan serve` для запуску серверу (все виконується в консолі OpenServer).

Для роботи з вебзастосуном для резервації та оренди житла користувач повинен у браузері на своєму пристрої ввести адресу вебсайту.

Б.5 Вхідні та вихідні дані

Вхідними даними вебзастосунку для резервації та оренди житла є інформація про користувача (логін, пароль), а також резервація житла, яку він виконує на початку роботи з застосунком.

Вихідними даними програмного забезпечення для організації та підтримки онлайн-чату є дані про його резервацію, що користувач отримує після завершення бронювання.

ДОДАТОК В
Текст програми

B.1 Зміст файлу Apartment.vue

```

<template>
<v-container
  grid-list-xs class="position-relative"
>
  <v-row>
    <v-col
      cols="12"
      sm="6"
      md="8"
    >
      <v-carousel>
        <v-carousel-item
          v-for="(image,i) in apartment.photos"
          :key="i"
          :src="'/img' + '/' + apartment.id + '/' + image.path"
          reverse-transition="fade-transition"
          transition="fade-transition"
        ></v-carousel-item>
      </v-carousel>
    </v-col>
    <v-col
      cols="6"
      md="4"
    >
      <v-card>
        <v-card-title>{{ apartment.name }}</v-card-title>
        <v-card-text>
          <v-row
            align="center"
            class="mx-0"
          >
            <v-rating
              :value="apartment.rating"
              color="amber"
              dense
              half-increments
              readonly
              size="14"
            ></v-rating>

            <div class="grey--text ms-4">

```

```

    {{ apartment.rating }} ({{ apartment.rating_count }})
  </div>
</v-row>

<div class="mt-4 mb-1">{{ apartment.price }} €</div>
<div class="my-1"><strong>Адреса:</strong>    {{ apartment.address
}}</div>
<span v-if="false">model: {{ dates }}</span>
<v-dialog
  ref="dialog"
  v-model="modal"
  :return-value.sync="date"
  persistent
  width="290px"
>
  <template v-slot:activator="{ on, attrs }">
    <v-text-field
      v-model="dateRangeText"
      label="Виберіть дати проживання"
      prepend-icon="mdi-calendar"
      readonly
      v-bind="attrs"
      v-on="on"
    ></v-text-field>
  </template>
  <v-date-picker
    v-model="dates"
    range
    scrollable
  >
    <v-spacer></v-spacer>
    <v-btn
      text
      color="primary"
      @click="modal = false"
    >
      Cancel
    </v-btn>
    <v-btn
      text
      color="primary"
      @click="$refs.dialog.save(date)"
    >
      OK

```

```

</v-btn>
</v-date-picker>
</v-dialog>

<div class="mt-4 mb-1"><strong>Загальна сума: {{ totalPrice }}
</strong></div>
</v-card-text>
<v-divider class="mx-4"></v-divider>
<v-card-actions>
<v-btn
color="deep-purple lighten-2"
text
@click="reserve(apartment.id)"
:disabled="!apartment.available"
>
Забронювати
</v-btn>
</v-card-actions>
</v-card>
</v-col>
</v-row>
<v-row>
<v-card
width="100%"
>
<v-card-text>
<div class="mt-2 mb-2">
<v-chip
v-for="(service,index) in apartment.services"
:key="index"
class="ma-2"
color="green"
outlined
>
<v-icon left>mdi-{{ service.icon}}</v-icon>
{{service.name}}
</v-chip>
</div>
<p>{{ apartment.description }}</p>
</v-card-text>
</v-card>
</v-row>
<v-row>
<v-card

```

```

width="100%"
>
<v-card-title>Відгуки</v-card-title>
<v-card-text>
<v-expansion-panels>
<v-expansion-panel
v-for="(rent,i) in apartment.rents"
:key="i"
>
<v-expansion-panel-header>
{{ rent.user_name }}
<v-rating
:value="rent.review_grade"
:color="amber"
dense
half-increments
readonly
size="14"
></v-rating>
</v-expansion-panel-header>
<v-expansion-panel-content>
{{ rent.review }}
</v-expansion-panel-content>
</v-expansion-panel>
</v-expansion-panels>
</v-card-text>
</v-card>
</v-row>
</v-container>
</template>

```

B.2 Зміст файлу ApartmentsList.vue

```

<template>
<v-row class="mb-6">
<v-col
v-for="(apartment,index) in apartments"
:key="apartment.id"
:cols="(12/itemsPerRow)"
class="py-2"
>
<v-card>
<v-img v-if="apartment.photos.length"

```

```

height="250"
:src="'/img' + '/' + apartment.id + '/' + apartment.photos[0].path"
></v-img>

<v-card-title>{{ apartment.name }}</v-card-title>
<v-card-text>
  <v-row
    align="center"
    class="mx-0"
  >
    <v-rating
      :value="apartment.rating"
      color="amber"
      dense
      half-increments
      readonly
      size="14"
    ></v-rating>
    <div class="grey--text ms-4">
      {{ apartment.rating }} ({{ apartment.rating_count }})
    </div>
  </v-row>
  <div class="mt-4 mb-1">{{ apartment.price }} €</div>
  <div class="my-1"><strong>Адреса:</strong> {{ apartment.address }}</div>
  <div class="my-1"><strong>Тип:</strong> {{ type(apartment.type) }}</div>
  <div>{{ apartment.short_description }}</div>
  <div class="mt-2">
    <v-btn
      v-for="(service,index) in apartment.services"
      :key="index"
      style="margin: 2px;"
      fab dark x-small color="green"
      :title="service.name"
    >
    <v-icon dark>mdi-{{ service.icon }}</v-icon>
  </v-btn>
</div>
</v-card-text>
<v-divider class="mx-4"></v-divider>
<v-card-actions>
  <v-btn
    :to="{ name: 'apartment', params: { apartmentId: apartment.id }}"
    color="deep-purple lighten-2"
  >

```

```

text
@click="reserve(apartment.id)"
:disabled="!apartment.available"
>
Зарезервувати
</v-btn>
</v-card-actions>
</v-card>
</v-col>
</v-row>
</template>

```

В.3 Зміст файлу BookingList.vue

```

<template>
<v-data-table
:headers="headers"
:items="rents"
sort-by="id"
class="elevation-1"
:footer-props="{
'items-per-page-options': [5,10,20,-1],
'items-per-page-all-text' : 'Всі',
'items-per-page-text': 'рядка на сторінку',
}"
>
<template v-slot:top>
<v-toolbar
flat
>
<v-toolbar-title>Історія бронювань</v-toolbar-title>
<v-dialog v-model="dialogRank" max-width="340px">
<v-card>
<v-card-title class="text-h5">Залиште відгук</v-card-title>
<v-card-text>
<v-rating
color="yellow darken-3"
background-color="grey darken-1"
empty-icon="$ratingFull"
half-increments
hover
large
></v-rating>

```

```

<v-textarea
solo
name="input-7-4"
label="Solo textarea"
></v-textarea>
</v-card-text>
<v-card-actions>
<v-spacer></v-spacer>
<v-btn color="blue darken-1" text @click="closeReviewPopup">Назад</v-
btn>
<v-btn color="blue darken-1" text @click="saveReview">Так</v-btn>
<v-spacer></v-spacer>
</v-card-actions>
</v-card>
</v-dialog>
<v-dialog v-model="dialogDelete" max-width="340px">
<v-card>
<v-card-title class="text-h5">Ви впевнені, що бажаєте скасувати
бронювання?</v-card-title>
<v-card-actions>
<v-spacer></v-spacer>
<v-btn color="blue darken-1" text @click="closeDelete">Назад</v-btn>
<v-btn
color="blue
darken-1 "
text
@click="deleteItemConfirm">Опублікувати</v-btn>
<v-spacer></v-spacer>
</v-card-actions>
</v-card>
</v-dialog>
</v-toolbar>
</template>
<template v-slot:item.apartment_name="{ item }">
<router-link :to="{ name: 'apartment', params: { apartmentId:
item.apartment_id } }">
{{ item.apartment_name }}
</router-link>
</template>
<template v-slot:item.status="{ item }">
<v-chip
v-if="item.status === 'cancel'"
class="ma-2"
color="red"
outlined
>Скасовано
</v-chip>

```

```

<v-chip
v-if="item.status === 'finish'"
class="ma-2"
color="green"
outlined
>Завершено
</v-chip>
<v-chip
v-if="item.status === 'process'"
class="ma-2"
color="blue"
outlined
>ВИКОНУЄТЬСЯ
</v-chip>
<v-chip
v-if="item.status === 'waiting'"
class="ma-2"
color="orange"
outlined
>в процесі підтвердження
</v-chip>
</template>
<template v-slot:item.paid="{ item }">
<v-icon
v-if="item.paid"
small
>mdi-check
</v-icon>
<v-icon
v-if="!item.paid"
small
>mdi-close
</v-icon>
</template>
<template v-slot:item.multiple="{ item }">
<v-icon
v-if="item.multiple"
small
>mdi-check
</v-icon>
<v-icon
v-if="!item.multiple"
small
>mdi-close

```

```

</v-icon>
</template>
<template v-slot:item.actions="{ item }">
  <v-icon
  small
  :disabled="item.status === 'waiting' || item.status === 'process' ||
item.review_grade !== ''"
  title="Залишити відгук"
  @click="addReview(item)"
  >
  mdi-plus
</v-icon>
<v-icon
  small
  :disabled="item.status !== 'waiting' && item.status !== 'process'"
  title="Скасувати бронювання"
  @click="deleteItem(item)"
  >
  mdi-delete
</v-icon>
</template>
</v-data-table>
</template>

```

В.4 Зміст файлу Faq.vue

```

<template>
<v-container
grid-list-xs class="position-relative"
>
<v-tabs
v-model="tab"
>
<v-tab>Оплата</v-tab>
<v-tab>Скасування</v-tab>
<v-tab>Типи номерів</v-tab>
</v-tabs>
<v-tabs-items v-model="tab">
<v-tab-item :key="0">
<v-expansion-panels>
<v-expansion-panel>
<v-expansion-panel-header>
Які засоби оплати ви приймаєте?

```

</v-expansion-panel-header>

<v-expansion-panel-content>

Банківська картка - найпоширеніший засіб оплати бронювань на Booking.com. Картки також використовують для підтвердження бронювань.

У деяких випадках приймаються альтернативні способи оплати, такі як PayPal. Іноді можна сплатити і дебетовою карткою.

Доступні способи оплати для обраного об'єкта будуть перелічені у підтвердженні бронювання. Ви також можете зателефонувати власникам об'єкта, щоб дізнатися, які варіанти оплати вони приймають.

</v-expansion-panel-content>

</v-expansion-panel>

<v-expansion-panel>

<v-expansion-panel-header>

Чи потрібно вносити заставу чи передоплату?

</v-expansion-panel-header>

<v-expansion-panel-content>

Деякі об'єкти вимагають внесення попередньої оплати або застави. Сума може становити повну вартість бронювання або її частину. Решта оплачується після заїзду.

Деякі об'єкти, однак, не вимагають передоплати, і сума бронювання списується повністю під час Вашого проживання в об'єкті. Вивчіть пункт про способи оплати у підтвердженні бронювання, щоб отримати докладнішу інформацію.

</v-expansion-panel-content>

</v-expansion-panel>

<v-expansion-panel>

<v-expansion-panel-header>

Чому я маю надавати дані своєї банківської картки?

</v-expansion-panel-header>

<v-expansion-panel-content>

Ці дані потрібні об'єкту розміщення, щоб підтвердити ваше бронювання. Адміністрація об'єкта може попередньо заблокувати* суму на вашій карті, щоб перевірити її дійсність та достатність коштів на рахунку або зняти оплату в момент бронювання.

*Попереднє блокування – це тимчасове блокування певної суми на карті, щоб перевірити її дійсність та достатність коштів на рахунку. Заблокована сума повертається на рахунок після певного проміжку часу – він залежить від об'єкта розміщення та вашого банку-емітента.

</v-expansion-panel-content>

</v-expansion-panel>

</v-expansion-panels>

</v-tab-item>

<v-tab-item :key="1">

<v-expansion-panels>

<v-expansion-panel>

<v-expansion-panel-header>

Чи можна скасувати бронювання?

</v-expansion-panel-header>

<v-expansion-panel-content>

Так! Штраф за відміну встановлює варіант розміщення, ви сплачуєте його безпосередньо. Умови вказані у правилах скасування.

</v-expansion-panel-content>

</v-expansion-panel>

<v-expansion-panel>

<v-expansion-panel-header>

Якщо мені потрібно скасувати бронювання, чи доведеться сплачувати штраф за його скасування?

</v-expansion-panel-header>

<v-expansion-panel-content>

Ви не платитимете штраф, якщо у вас бронювання з безкоштовним скасуванням. Якщо воно безповоротне або його більше не можна скасувати безкоштовно, можливо, доведеться сплатити штраф. Сума встановлюється варіантом розміщення та виплачується безпосередньо.

</v-expansion-panel-content>

</v-expansion-panel>

<v-expansion-panel>

<v-expansion-panel-header>

Чи можна змінити дати у бронюванні за безповоротним тарифом?

</v-expansion-panel-header>

<v-expansion-panel-content>

За скасування бронювання за безповоротним тарифом зазвичай береться штраф. Однак ви можете запросити безкоштовне скасування у розділі управління бронюванням. Ваш запит буде відправлено до об'єкта розміщення, в якому можуть вирішити не брати з вас плату за скасування. Зміна дат бронювання за безповоротним тарифом неможлива. Однак, якщо з вас не стягуватимуть штрафу за скасування, ви можете знову оформити бронювання на бажані дати.

</v-expansion-panel-content>

</v-expansion-panel>

<v-expansion-panel>

<v-expansion-panel-header>

Як дізнатися, що моє бронювання було скасовано?

</v-expansion-panel-header>

<v-expansion-panel-content>

Ви повинні отримати лист із підтвердженням скасування. Перевірте пошту та папку «Спам». Якщо ви не отримаєте підтвердження протягом 24 годин, зв'яжіться з об'єктом розміщення, щоб уточнити, чи вони отримали запит на скасування.

</v-expansion-panel-content>
 </v-expansion-panel>
 </v-expansion-panels>
 </v-tab-item>

<v-tab-item :key="2">

<v-expansion-panels>

<v-expansion-panel>

<v-expansion-panel-header>

У чому різниця між двомісним номером та номером Твін?

</v-expansion-panel-header>

<v-expansion-panel-content>

У двомісному номері встановлюється 1 двоспальне ліжко, а в номері Твін - 2 односпальні ліжка. У двомісному номері/номері Твін можуть бути встановлені ліжка обох типів. Ви можете вказати кращий тип ліжка у полі "Особливі побажання" при бронюванні.

</v-expansion-panel-content>

</v-expansion-panel>

<v-expansion-panel>

<v-expansion-panel-header>

Що означає «вартість не повертається» та «безкоштовне скасування»?

</v-expansion-panel-header>

<v-expansion-panel-content>

Кожен номер (варіант житла) має свої правила бронювання, які встановлює об'єкт розміщення.

«Вартість не повертається» означає, що якщо ви вирішите скасувати або змінити бронювання, стягуватиметься штраф. Розмір штрафу зазначений у правилах у процесі бронювання та у підтвердженні бронювання.

«Безкоштовне скасування» означає, що ви можете скасувати або змінити бронювання без штрафу протягом терміну, встановленого об'єктом розміщення (наприклад, «Безкоштовне скасування протягом N днів до заїзду» або «Безкоштовне скасування до дд/мм/гг чч:мм»). Ця інформація зазначена у правилах у процесі бронювання та у підтвердженні бронювання.

</v-expansion-panel-content>

</v-expansion-panel>

<v-expansion-panel>

<v-expansion-panel-header>

Чи можу я попросити встановити додаткове ліжко в номері, і чи стягується за це додаткова плата?

</v-expansion-panel-header>

<v-expansion-panel-content>

Інформація про додаткові ліжка міститься в розділі «Умови розміщення» на сторінці об'єкта розміщення.

Додаткова плата (якщо плата стягується готелем) не включена у вартість бронювання.

Щоб запросити додаткове ліжко в процесі бронювання, використовуйте поле «Розкажіть нам про ваші особливі побажання».

Якщо ви вже зробили бронювання, ви можете запросити додаткове ліжко, пройшовши посилання в електронному листі з підтвердженням бронювання.

Радимо зв'язатись з об'єктом розміщення до приїзду для гарантії виконання вашого запиту. Контактні дані об'єкта можна переглянути у вашому електронному листі з підтвердженням бронювання або на сторінці бронювання у вашому обліковому записі.

```
</v-expansion-panel-content>
</v-expansion-panel>
</v-expansion-panels>
</v-tab-item>
</v-tabs-items>
</v-container>
</template>
```

В.5 Зміст файлу Settings.vue

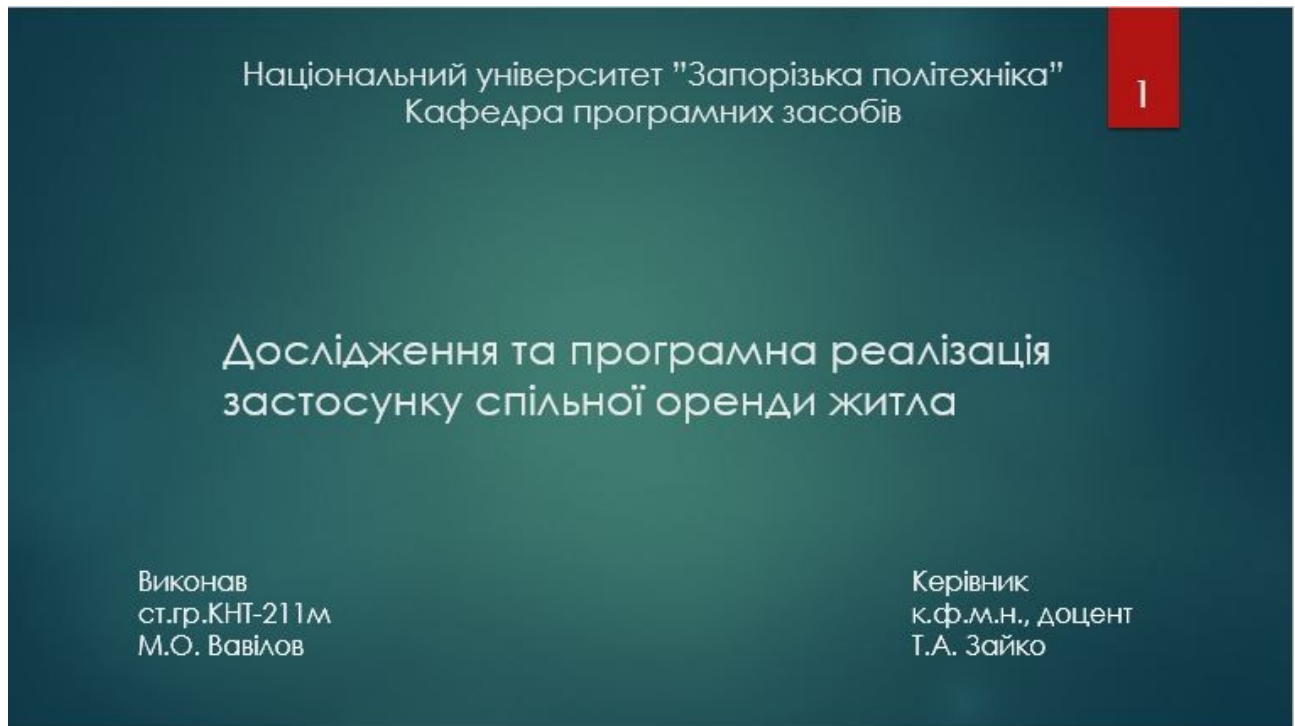
```
<template>
<v-container>
<v-row align="center">
<v-col cols="12" sm="2">
</v-col>
</v-row>
<v-form
ref="form"
v-model="valid"
lazy-validation
>
<v-text-field
v-model="user.name"
:counter="10"
:rules="nameRules"
label="Ім'я"
required
></v-text-field>
<v-text-field
v-model="user.email"
label="Адреса електронної пошти"
disabled
></v-text-field>
<v-text-field
```

```

v-model="user.phone1"
:rules="phoneRules"
label="Телефон №1"
type="number"
required
></v-text-field>
<v-text-field
v-model="user.phone2"
:rules="phoneRules"
label="Телефон №2"
></v-text-field>
<v-select
v-model="user.country_code"
:items="items"
:rules="[v => !!v || 'Item is required']"
label="Країна"
required
></v-select>
<v-btn
:disabled="!valid"
color="success"
class="mr-4"
@click="validate"
>
Зберегти
</v-btn>
</v-form>
</v-container>
</template>

```

ДОДАТОК Г
Слайди презентації



Національний університет "Запорізька політехніка"
Кафедра програмних засобів

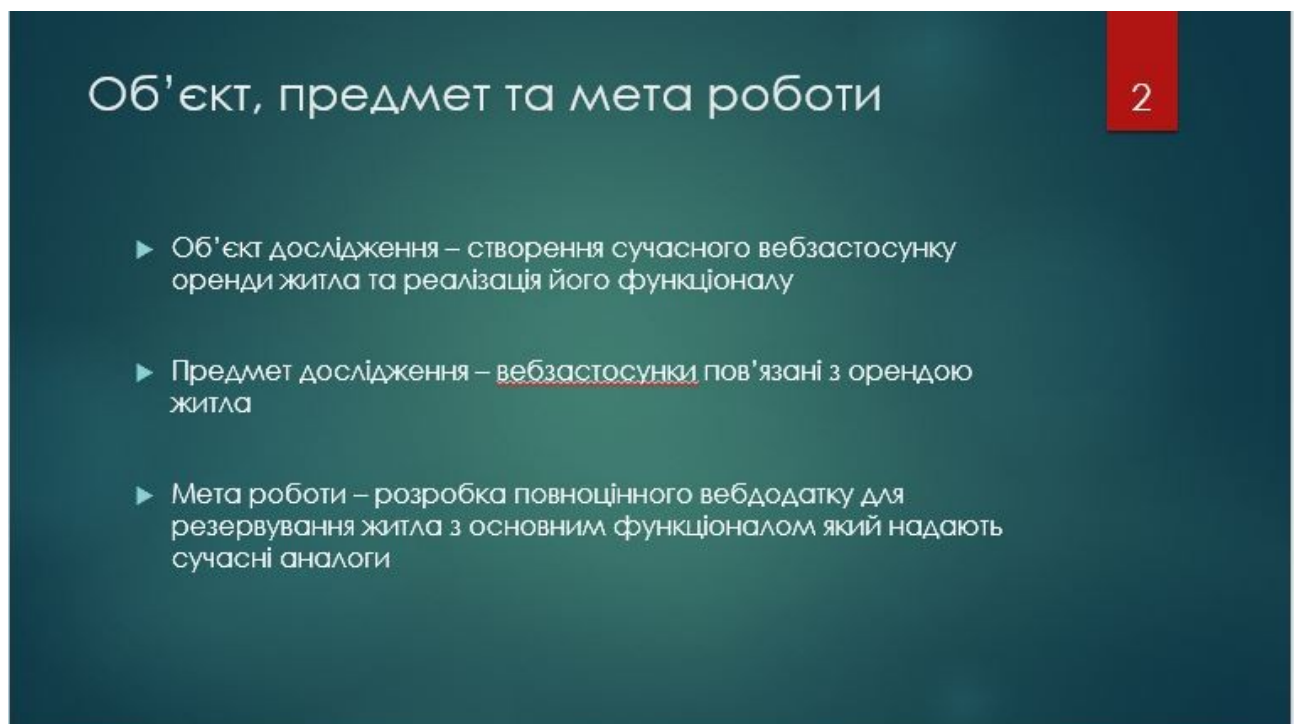
1

Дослідження та програмна реалізація
застосунку спільної оренди житла

Виконав
ст.гр.КНТ-211м
М.О. Вавілов

Керівник
к.ф.м.н., доцент
Т.А. Зайко

Рисунок Г.1 – Слайд №1



Об'єкт, предмет та мета роботи

2

- ▶ Об'єкт дослідження – створення сучасного вебзастосунку оренди житла та реалізація його функціоналу
- ▶ Предмет дослідження – вебзастосунки пов'язані з орендою житла
- ▶ Мета роботи – розробка повноцінного вебдодатку для резервування житла з основним функціоналом який надають сучасні аналоги

Рисунок Г.2 – Слайд №2

Завдання роботи

3

- ✓ Аналіз предметної області та аналогів
- ✓ Аналіз необхідних технологій
- ✓ Вибір мов та фреймворків бекенду та фронтенду
- ✓ Діаграми взаємодій та дизайн вебдодатку
- ✓ Структура проекту та база даних
- ✓ Налаштування та підключення OpenServer
- ✓ Перевірка працездатності вебзастосунку
- ✓ Результати та висновки

Рисунок Г.3 – Слайд №3

Порівняння існуючих систем букінгу

4

Функціонал	Booking	Agoda	Expedia	Hotellook
Оренда машини, таксі, бронювання білетів на літак	Так	Так	Так	Ні
Широкий вибір валюти та мови	Так	Так	Так	Так
Порівняння цін з іншими системами	Ні	Ні	Ні	Так
Пропозиції для подорожей з фільтром по бюджету	Так	Ні	Ні	Ні
Автоматичне виправлення запитів при пошуку	Так	Так	Ні	Так

Рисунок Г.4 – Слайд №4

Технології необхідні для розробки

5

- ✓ Мова гіпертекстової розмітки HTML
- ✓ Мова опису стилів CSS
- ✓ Динамічна об'єктно-орієнтована мова програмування JavaScript
- ✓ Мова програмування PHP

Критерії порівняння	Мови програмування		
	PHP	JavaScript	Python
Ознайомленість та швидкість роботи	+ -	+	-
Гнучкість та ефективність під вебзастосунок для оренди житла	+ -	+	+ -
Легкість вивчення	+ -	+ -	+ -

Рисунок Г.5 – Слайд №5

Програмні засоби

6

Фронтенд фреймворком біло обрано Vue.js який є прогресивним, повністю пригодний для створення складних одно сторінкових вебдодатків(SPA) й в порівнянні з аналогами має найкращу документацію, бекенд фреймворком було обрано Laravel який має перевагу над аналогами в продуманій архітектурі з простою документацією, надійна база даних, постійні оновлення та гнучка адміністративна панель, обидва мають безкоштовну модель розповсюдження.

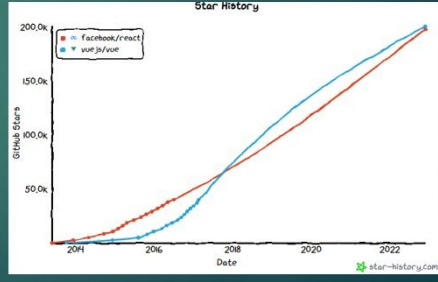
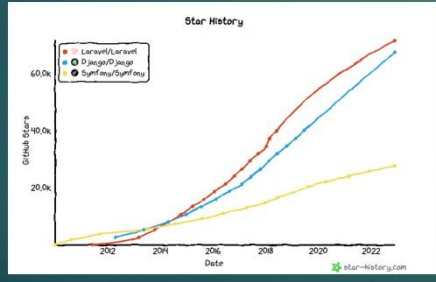


Рисунок Г.6 – Слайд №6

Редактор коду

7

Редактором коду було обрано безкоштовний Visual Studio Code.

Переваги:

- ✓ Безкоштовна модель розповсюдження
- ✓ Вбудований термінал
- ✓ Підтримка Git
- ✓ Невеликий розмір
- ✓ Велика кількість корисних розширень
- ✓ Підтримка більшості сучасних технологій розробки

Критерії порівняння	Visual Studio Code	PHPStorm	Sublime Text
Мала витрати ресурсів	+/-	-	+
Безкоштовний	+	-	+/-
Відладка коду	+	+	-
Установка розширень	+	+/-	+/-

Рисунок Г.7 – Слайд №7

SPA – single page application

8

SPA – також відомий як односторінковий інтерфейс, вебзастосунок або вебсайт який вміщується на одній сторінці зменшуючи навантаження на сервер, та інтернет. Був обраний для як найбільш актуальний для даного вебзастосунку.

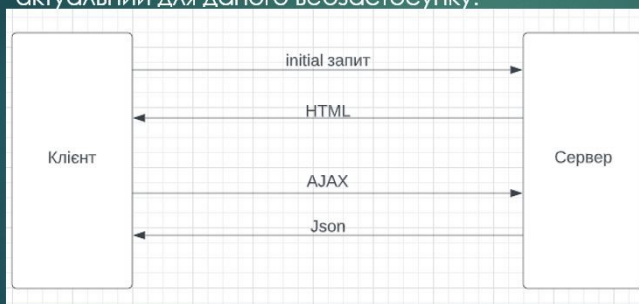


Рисунок Г.8 – Слайд №8

Логіка запитів

9

Вся логіка запитів в SPA-вебзастосунку організована через перший initial запит, який повертає html `app.blade.php`.

А всі послідуючі це AJAX:

- ❖ Get `/api/apartments`
- ❖ Get `/api/apartment/1`
- ❖ Post `/api/apartment/1/rent`
- ❖ Get `/api/rents`
- ❖ Delete `/api/rent/1`
- ❖ Post `/api/review`

Рисунок Г.9 – Слайд №9

Шаблон та компоненти

10

Шаблон blade зберігає скрипт ініціалізації застосунку на фреймворк vue. Є єдиним запитом, який повертає html, і є входною точкою в застосунку. Після чого йдуть лише AJAX запити.

Blade шаблон

- auth
- layouts
- app.blade.php
- home.blade.php
- welcome.blade.php

КОМПОНЕНТИ

- Dashboard
- Apartment.vue
- ApartmentBooking.vue
- ApartmentList.vue
- BookingList.vue
- Bookings.vue
- Contacts.vue
- Faq.vue
- Settings.vue

Рисунок Г.10 – Слайд №10

Можливості вебзастосунок

11

Розроблений вебзастосунок надає можливість:

- ✓ Обирати та резервувати житло
- ✓ Перевіряти статус бронювання
- ✓ Залишати та читати відгуки користувачів та орендаторів

Рисунок Г.11 – Слайд №11

Діаграми активності взаємодії клієнта і сервера а також діаграма дій у системі

12

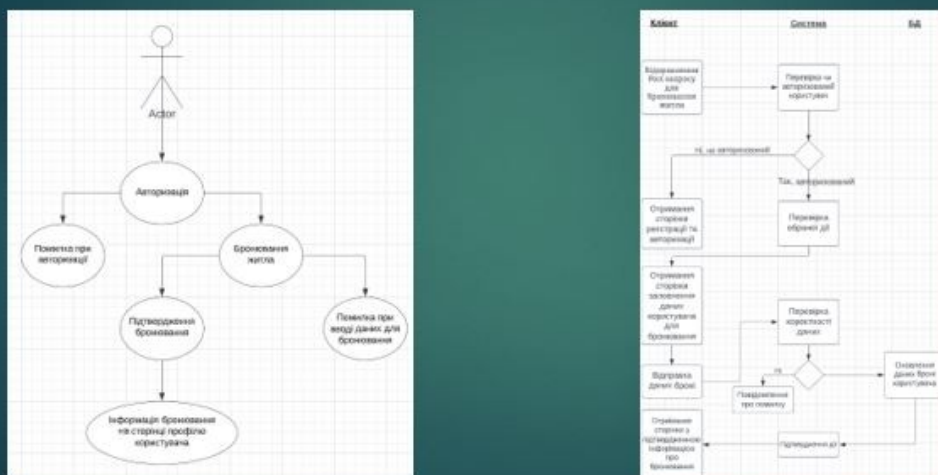


Рисунок Г.12 – Слайд №12

Результати роботи. Головна сторінка застосунку

13

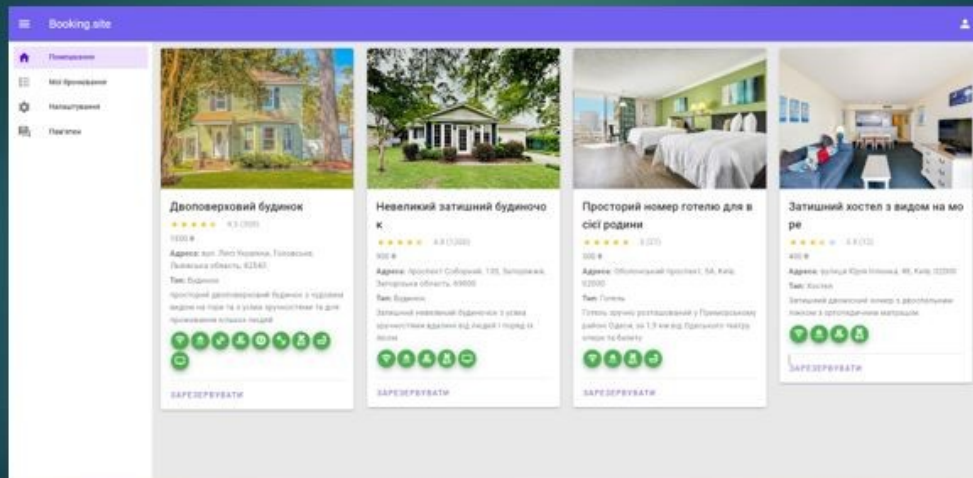


Рисунок Г.13 – Слайд №13

Результати роботи. Сторінка Резервації

14

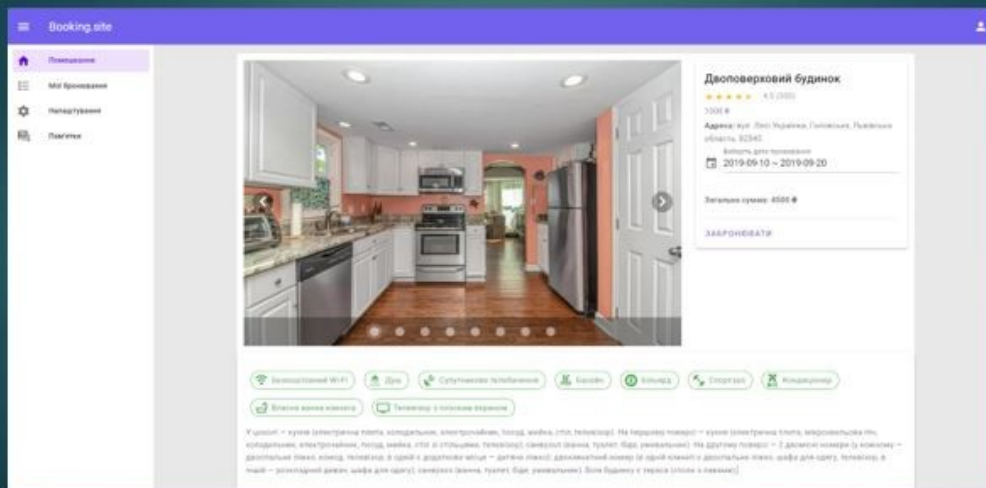
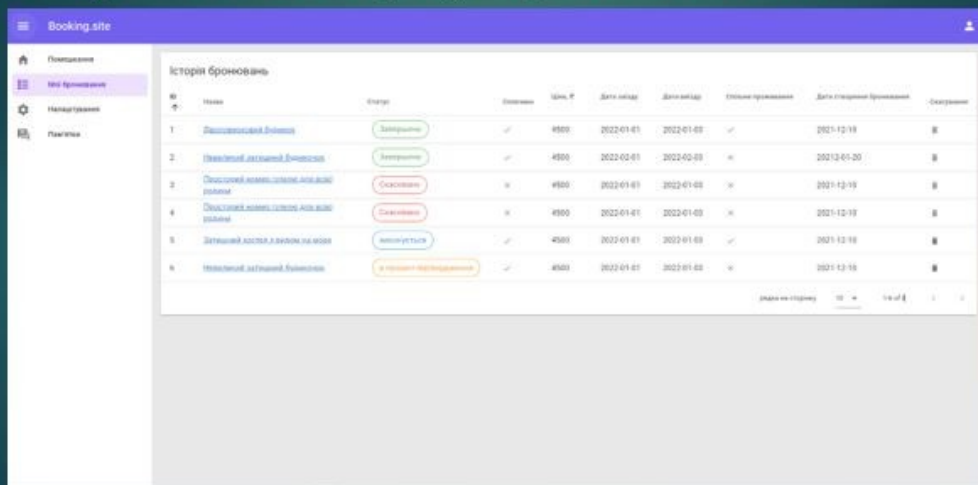


Рисунок Г.14 – Слайд №14

Результати роботи. Сторінка статусу бронювання

15

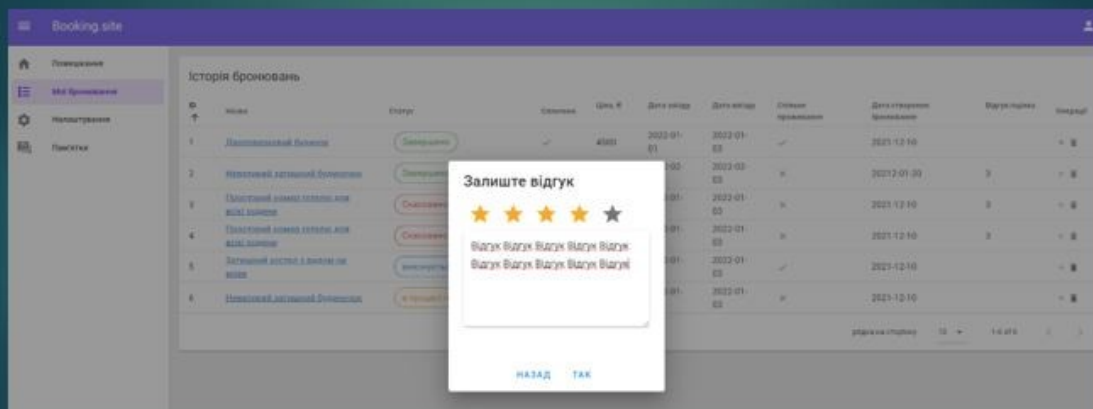


№	Назва	Статус	Скасування	Ціна, ₴	Дата виїзду	Дата заїзду	Статус бронювання	Дата створення бронювання	Скасування
1	Домашній бізнес	Затверджено	✓	4500	2022-01-01	2022-01-03	✓	2021-12-19	✕
2	Домашній бізнес	Затверджено	✓	4500	2022-02-01	2022-02-03	✕	2021-01-20	✕
3	Домашній бізнес	Скасовано	✕	4500	2022-01-01	2022-01-03	✕	2021-12-19	✕
4	Домашній бізнес	Скасовано	✕	4500	2022-01-01	2022-01-03	✕	2021-12-19	✕
5	Домашній бізнес	Виконано	✓	4500	2022-01-01	2022-01-03	✓	2021-12-19	✕
6	Домашній бізнес	В процесі бронювання	✓	4500	2022-01-01	2022-01-03	✕	2021-12-19	✕

Рисунок Г.15 – Слайд №15

Результати роботи. Відгук

16



№	Назва	Статус	Скасування	Ціна, ₴	Дата виїзду	Дата заїзду	Статус бронювання	Дата створення бронювання	Відгук	Відгук
1	Домашній бізнес	Затверджено	✓	4500	2022-01-01	2022-01-03	✓	2021-12-19	✕	✕
2	Домашній бізнес	Затверджено	✓	4500	2022-02-01	2022-02-03	✕	2021-01-20	✕	✕
3	Домашній бізнес	Скасовано	✕	4500	2022-01-01	2022-01-03	✕	2021-12-19	✕	✕
4	Домашній бізнес	Скасовано	✕	4500	2022-01-01	2022-01-03	✕	2021-12-19	✕	✕
5	Домашній бізнес	Виконано	✓	4500	2022-01-01	2022-01-03	✓	2021-12-19	✕	✕
6	Домашній бізнес	В процесі бронювання	✓	4500	2022-01-01	2022-01-03	✕	2021-12-19	✕	✕

Рисунок Г.16 – Слайд №16

Результати роботи. Сторінка налаштувань користувача

17

Booking site

Повіщення
Мій бронювання
Налаштування
Помітки

Ім'я
John Smith

Електронна пошта
john.smith@gmail.com

Телефон 01
0661234567

Телефон 02
0679876543

Країна
Україна

Зберегти

Рисунок Г.17 – Слайд №17

Результати роботи. Сторінка пам'ятки

18

Booking site

Повіщення
Мій бронювання
Налаштування
Помітки

ОПЛАТА СХАСТУВАННЯ ІНШІ КОМЕНТИ

Ваше запитання про оплату?

Чи потрібно мені внести заставу чи передоплату?

Чому я не можу надати дані своєї банківської картки?

Рисунок Г.18 – Слайд №18

Результати роботи. Сторінка реєстрації

19

Регістр. Зареєструватися

Регістрація

Ім'я

Адреса електронної пошти

Пароль

Підтвердіть пароль

Рисунок Г.19 – Слайд №19

Результати роботи. Сторінка авторизації

20

Регістр. Зареєструватися

Авторизація

Адреса електронної пошти

Пароль

☐ Запам'ятати мене

[Забудьте свій пароль?](#)

Рисунок Г.20 – Слайд №20

Висновки

21

У результаті проведеної роботи за допомогою фреймворків Laravel для бекенду та Vue.js для фронтенду, було створено SPA вебзастосунок, який дає можливість орендувати житло в пару кліків з можливістю відстеження статусу резервування.

В порівнянні з аналогами які на ринку вже більше двох десятиліть, є можливості для модифікації застосунку:

- ✓ Додавання геолокації користувача для пошуку найближчих варіантів оренди
- ✓ Онлайн чат-підтримка

Рисунок Г.21 – Слайд №21

22

Дякую за увагу!

Рисунок Г.22 – Слайд №22