**1 ВСТУП**

1.1 Огляд продукту

Створена серверна система керування бронюваннями ресурсів автоматично обробляє запити на резервування, спираючись на актуальні дані про доступність об’єктів, задані правила використання та профілі користувачів. Платформа підтримує багаторівневу авторизацію, зручне адміністрування користувачів і ресурсів, зберігає всі операції у централізованій базі даних із функцією резервного копіювання та безболісно масштабується зі зростанням навантажень.

1.2 Мета

Метою створюваного веб-сервісу Booking є повна автоматизація процесів пошуку, бронювання й адміністрування ресурсів. Система дає змогу в реальному часі резервувати помешкання чи інші об’єкти, мінімізує ручні помилки, оптимізує заповнюваність та забезпечує прозору аналітику для власників, водночас залишаючись максимально зручною для кінцевих користувачів.

1.3 Межі

Серверна частина:

* реалізована на основі .NET 3.1 (ASP.NET Core) з REST-архітектурою; HTTP-API обслуговує web-клієнт, мобільний застосунок та адмін-панель;
* дані про нерухомість, бронювання та користувачів зберігаються у MySQL; БД нормалізована й має реплікацію для читальних навантажень;
* роль-орієнтована авторизація з JWT-токенами та контрольованим часом дії;
* гарантована коректність одночасних операцій бронювання (ACID) і push-сповіщення клієнтам через черги повідомлень;
* залежності керуються через NuGet; збірка й запуск здійснюються .NET CLI.

Інтеграції та зовнішні служби:

* платіжні шлюзи для обробки транзакцій;
* картографічний сервіс для відображення локацій;
* email- та SMS-сповіщення;
* web-хуки для синхронізації доступності з агрегаторами.

Функціональні обмеження

* користувач може мати лише одне активне бронювання на задані дати;
* доступ до персональних даних і статистики можливий лише після аунтифікації;
* e-mail-логін повинен бути унікальним;
* офіційно підтримуються українська та англійська локалізації;
* усі з’єднання шифруються.

1.4 Посилання

В проєкті були використані такі посилання:

* Сервіс з бронювання житла : веб-сайт. — Режим доступу: <https://www.airbnb.com> (дата звернення: 03.05.2025).
* Krug S. *Don’t Make Me Think: A Common-Sense Approach to Web Usability*. — Berkeley : New Riders, 2005. — 201 с.
* SQL Stored Procedures : веб-сайт навчального порталу W3Schools. — Режим доступу: <https://www.w3schools.com/sql/sql_stored_procedures.asp> (дата звернення: 04.05.2025).
* Create Data Transfer Objects (DTOs) : документація Microsoft ASP.NET Web API. — Режим доступу: <https://docs.microsoft.com/en-us/aspnet/web-api/overview/data/using-web-api-with-entity-framework/part-5> (дата звернення: 04.05.2025).
* Model validation in ASP.NET Core MVC and Razor Pages : документація Microsoft. — Режим доступу: <https://docs.microsoft.com/aspnet/core/mvc/models/validation?view=aspnetcore-3.1> (дата звернення: 05.05.2025).
* ControllerBase.BadRequest Method : документація Microsoft. — Режим доступу: <https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.mvc.controllerbase.badrequest?view=aspnetcore-3.1> (дата звернення: 05.05.2025).
* Об’єктно-реляційне відображення : стаття в *Вікіпедії*. — Режим доступу: [https://uk.wikipedia.org/wiki/Об’єктно-реляційне\_відображення](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%E2%80%99%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D1%96%D0%B9%D0%BD%D0%B5_%D0%B2%D1%96%D0%B4%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F) (дата звернення: 06.05.2025).
* Лазаро Іссі Коен, Джозеф Іссі Коен. *Повний довідник по HTML, CSS і JavaScript*. — К. : ЕКОМ Паблішерз, 2016. — 311 с.
* Херман Д. *Сила JavaScript. 68 способів ефективного використання JS*. — М. : Пітер, 2015. — 952 с.
* Кингслі-Х'ю К. Е. *JavaScript 1.5: учебный курс*. — М. : Вильямс, 2013. — 272 с.
* Крокфорд Д. *JavaScript. Сильні сторони*. — М. : Пітер, 2016. — 262 с.

1.5 Означення та абревіатури

|  |  |
| --- | --- |
| Термін | Оновлене визначення |
| Користувач | Особа, що взаємодіє із сервісом бронювання через веб-сайт або мобільний застосунок. Ролі:   * гість; * агент/власник; * адміністратор. |
| Сервер | ASP.NET Core веб-API, що приймає запити клієнтів, виконує бізнес-логіку (пошук, розрахунок вартості, перевірка доступності) та зберігає дані у базі. |
| Оголошення | Запис у системі, який описує об’єкт нерухомості: назву, локацію, фото, правила проживання, ціну тощо. |
| Бронювання | Транзакційна операція, що фіксує період оренди житла гостьовою стороною та гарантує недосяжність дат для інших користувачів. |

2 ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

* розширення інтелектуального пошуку та фільтрів, зокрема рекомендацій на основі історії переглядів і геоконтекстних запитів;
* поліпшення алгоритмів динамічного ціноутворення для більш точного прогнозування попиту та прозорого формування вартості;
* інтеграція розумних пристроїв у помешкання, наприклад електронних замків і датчиків енергоспоживання, для безконтактного заселення й віддаленого контролю;
* розроблення повноцінного мобільного застосунку з офлайн-доступом до квитів та push-нагадуваннями;
* перехід на хмарну багатотенантну архітектуру з автоматичним масштабуванням для мереж керуваних апартаментів.

2.2 Функції продукту

Користувачі отримують такі можливості:

* реєстрація та аутентифікація;
* перегляд і керування власними оголошеннями та бронюваннями;
* перегляд списку користувачів і надання ролей із подальшим скасуванням за потреби;
* імпорт та експорт даних у форматі CSV;
* CRUD-операції для оголошень, бронювань і календарів доступності;
* перегляд фінансових звітів і журналів дій.

2.3 Характеристики користувачів

Адміністратори:

* керують користувачами, ролями та системними налаштуваннями;
* мають підготовку в галузі безпеки й даних;
* працюють із великими таблицями експорту й імпорту.

Власники або агенти:

* додають і редагують оголошення, ціни та календарі;
* використовують базові інструменти аналітики та повідомлення про бронювання;
* очікують зрозумілий інтерфейс керування нерухомістю.

Гості:

* шукають та бронюють житло;
* цінують зрозумілий процес оплати й швидкі підтвердження;
* можуть не мати технічних навичок, проте потребують мобільності та чітких сповіщень.

2.4 Загальні обмеження

* поки що доступний лише веб-інтерфейс, мобільний застосунок у розробленні;
* доступ до ресурсів визначено за ролями;
* підтримуються українська та англійська мови.

2.5 Припущення та залежності

* необхідне стабільне інтернет-з’єднання для обробки запитів і платежів;
* сервер працює на платформі .NET 8, база даних MySQL;
* зовнішні платіжні шлюзи викликаються через REST API;
* робота динамічного ціноутворення залежить від актуальних даних про попит і конкуренцію.

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

* веб-застосунок, що працює у сучасних браузерах;
* мобільний клієнт iOS та Android у дорожній карті;
* Swagger UI або Postman-колекція для розробників;
* ресурси API: реєстрація, автентифікація, оголошення, бронювання, користувачі, ролі, платежі;
* мінімум полів у формах, підказки та валідація в реальному часі.

3.1.2 Апаратний інтерфейс

* клієнт: підключення до Інтернету, двоядерний процесор, 2 ГБ RAM, 100 МБ вільного місця;
* сервер: дві віртуальні CPU, 4 ГБ RAM, 20 ГБ SSD, канал 100 Мбіт с.

3.1.3 Програмний інтерфейс

* бекенд ASP.NET Core 8, Entity Framework Core, MySQL;
* RESTful API з обміном JSON.

3.1.4 Комунікаційний протокол

* усі запити через HTTPS;
* автентифікаційний токен JWT у заголовку Authorization.

3.1.5 Обмеження пам’яті

* клієнт утримує до 100 МБ кешу;
* сервер резервує до 10 ГБ під базу даних при одному мільйоні записів.

3.1.6 Операції

* створення акаунта і вхід;
* пошук та перегляд житла;
* створення і скасування бронювань;
* CRUD операції для оголошень, календарів, ролей;
* експорт і імпорт даних CSV;
* перегляд фінансових транзакцій та журналів дій.

3.1.7 Функції продукту

* розширений пошук і фільтри;
* безпечне бронювання з онлайн-оплатою;
* повідомлення електронною поштою та SMS;
* аналітика для власників;
* двомовний інтерфейс.

3.1.8 Припущення і залежності

* користувачі мають стабільний інтернет;
* платіжні шлюзи та картографічні сервіси доступні;
* система покладається на бібліотеки .NET і NuGet;
* безпека гарантується HTTPS та JWT.

3.2 Властивості програмного продукту

* управління оголошеннями та календарем доступності;
* обробка платежів і повернень;
* онлайн-чат між гостем та власником;
* резервне копіювання й відновлення бази;
* аналітичні панелі для відстеження доходу та заповнюваності.

3.3 Атрибути програмного продукту

3.3.1 Надійність

* час простою не перевищує нуль цілих одна десята відсотка на місяць.

3.3.2 Доступність

* доступність сервісу щонайменше дев’яносто дев’ять цілих дев’ять десятих відсотка на місяць;
* підтримка будь-якого пристрою з браузером.

3.3.3 Безпека

* паролі зберігаються як bcrypt-хеші;
* автентифікація через JWT;
* журнали безпеки зберігаються п’ять років.

3.3.4 Супроводжуваність

* покриття юніт-тестами шістдесят відсотків;
* стандарт кодування і автоматичний аналіз стилю.

3.3.5 Переносимість

* контейнер Docker запускається однією командою;
* відсутність залежності від платформи Windows чи специфічних бібліотек.

3.3.6 Продуктивність

* час відповіді пошуку менше однієї секунди;
* підтримка тисячі запитів на хвилину на одному екземплярі сервера.

3.4 Вимоги до бази даних

* реляційна MySQL або PostgreSQL;
* таблиці Users, Roles, Listings, Bookings, Payments, Logs;
* індекси по зовнішніх та первинних ключах;
* резервні копії зберігаються тридцять днів.

3.5 Інші вимоги

* повна локалізація українською та англійською;
* відповідність WCAG рівень AA.