

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

«Платформа для проведення аукціонів»

Виконав студент групи ІПЗ-21-5
ВОЛКОВ Олександр Максимович

Керівник роботи:
САВІЦЬКИЙ Роман Святославович

Рецензент:
ВЛАСЕНКО Олег Васильович

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна Вакалюк

«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ВОЛКОВ Олександр Максимович**

Керівник роботи: **САВІЦЬКИЙ Роман Святославович**

Тема роботи: **«Платформа для проведення аукціонів»**,

затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**

Вихідні дані для роботи: Об'єктом дослідження є використання інформаційних технологій для автоматизації процесів проведення онлайн аукціонів.

Предметом дослідження є використання технологій розробки вебзастосунків для автоматизації проведення онлайн аукціонів.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Савіцький Р.С.	27.02.2025р.	27.02.2025р.
2	Савіцький Р.С.	27.02.2025р.	27.02.2025р.
3	Савіцький Р.С.	27.02.2025р.	27.02.2025р.

РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня «бакалавр» складається з вебдодатку для проведення аукціонів та пояснювальної записки. Пояснювальна записка до кваліфікаційної роботи містить 64 сторінки, 40 ілюстрацій та 18 таблиць. Загальний обсяг 67 сторінок.

Метою роботи є створення платформи для проведення аукціонів з використанням сучасних технологій веброзробки. Система включає модулі для управління лотами, аукціонами, користувачами, замовленнями.

У процесі розробки було здійснено порівняльний аналіз схожих програмних продуктів. Представлено основні сценарії використання системи, логіку роботи та алгоритми ключових функцій, а також розроблено базу даних з використанням MongoDB.

Технологічний стек проекту включає сучасні інструменти веб-розробки. Клієнтська частина побудована на Next.js і React з використанням React Hook Form, Tanstack React Query, Radix UI та Tailwind CSS. Серверна частина інтегрована в Next.js через API Routes, з ORM Prisma для роботи з базою даних, NextAuth для автентифікації, bcrypt для захисту паролів, Stripe для платежів та Socket.io для комунікації в реальному часі. Архітектура системи базується на клієнт-серверній взаємодії та включає модулі управління лотами, проведення аукціонів, адміністрування та оформлення замовлень.

КЛЮЧОВІ СЛОВА: АУКЦІОН, ЛОТ, ПЛАТФОРМА, СТАВКА, ТОРГИ.

					ІПЗ.КР.Б – 121 – 25 – ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Волков О.М.			Платформа для проведення аукціонів Пояснювальна записка			Літ.		Арк.	Аркушів	
Керівник		Савіцький Р.С.								3	67	
								Житомирська політехніка, група ІПЗ-21-5				
Рецензент		Власенко О.В.										
Зав. каф.		Вакалюк Т.А.										

ABSTRACT

The qualification work for the bachelor's degree consists of a web application for conducting auctions and an explanatory note. The explanatory note to the qualification work contains 64 pages, 40 illustrations and 18 tables. The total volume is 67 pages.

The purpose of the work is to create a platform for conducting auctions using modern web development technologies. The system includes modules for managing lots, auctions, users, and orders.

During the development process, a comparative analysis of similar systems was carried out. The main scenarios for using the system, the logic of operation and algorithms of key functions were presented, and a database was developed using MongoDB.

The technological stack of the project includes modern web development tools. The client part is built on Next.js and React using React Hook Form, Tanstack React Query, Radix UI and Tailwind CSS. The server side is integrated into Next.js via the Routes API, with the Prisma ORM for database management, NextAuth for authentication, bcrypt for password protection, Stripe for payments, and Socket.io for real-time communication. The system architecture is based on client-server interaction and includes modules for lot management, auctions, administration, and order processing.

KEYWORDS: AUCTION, LOT, PLATFORM, BID, TRADING.

					ИПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						4

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ОРГАНІЗАЦІЇ ТА УПРАВЛІННЯ ПЛАТФОРМОЮ ДЛЯ ПРОВЕДЕННЯ АУКЦІОНІВ.....	9
1.1 Постановка задачі	9
1.2 Аналіз існуючих рішень.....	11
1.3 Вибір архітектури	14
1.4 Обґрунтування вибору інструментальних засобів.....	15
Висновок до першого розділу	17
РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАТФОРМИ ДЛЯ ПРОВЕДЕННЯ АУКЦІОНІВ	19
2.1 Визначення варіантів використання та об’єктно-орієнтованої структури системи	19
2.2 Розробка бази даних системи	24
2.3 Проектування та реалізація алгоритмів роботи системи.....	30
2.4 Реалізація системи аукціонних торгів	36
Висновок до другого розділу	40
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ ДЛЯ ПРОВЕДЕННЯ АУКЦІОНІВ.....	42
3.1 Порядок встановлення та налаштування параметрів системи.....	42
3.2 Структура інтерфейсу платформи для проведення аукціону	44
3.3 Тестування роботи програмного комплексу.....	56
Висновок до третього розділу	60
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТКИ.....	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – База Даних

API – Application Programming Interface

JSON – JavaScript Object Notation

ORM – Object-Relational Mapping

REST API – Representational State Transfer Application Programming Interface

SEO – Search Engine Optimization

SQL – Structured Query Language

SSG – Static Site Generation

SSR – Server Site Rendering

UI – User Interface

ВСТУП

Актуальність теми. Онлайн-аукціони є одним із найдинамічніших напрямів електронної комерції, забезпечуючи прозорість, доступність і ефективність торгівлі товарами та послугами. Платформи для проведення аукціонів відкривають широкі можливості для продавців і покупців, дозволяючи залучати більшу кількість учасників, знижувати витрати на організацію фізичних торгів і спрощувати процес купівлі-продажу.

Світовий ринок онлайн-аукціонів демонструє стабільне зростання протягом останніх років. Особливо актуальними стали онлайн-аукціони в умовах військової агресії, коли фізична присутність на торгах є обмеженою, що підкреслило необхідність розвитку надійних цифрових платформ для проведення аукціонів.

Розвиток технологій, зокрема штучного інтелекту, криптографічних методів захисту, хмарних технологій, підвищує безпеку, довіру та ефективність аукціонних платформ. Інтеграція із сучасними платіжними системами дозволяє швидко та безпечно здійснювати фінансові транзакції.

Таким чином, створення платформи для проведення аукціонів є актуальним напрямом, що сприяє цифровій трансформації торгівлі, розширенню ринку та підвищенню конкурентоспроможності бізнесу.

Метою випускної кваліфікаційної роботи є розробка алгоритмів роботи та реалізація веборієнтованої платформи для проведення аукціонів, що забезпечує прозорість, безпеку та ефективність торгів.

Встановлена мета обумовлює наступні завдання:

1. Проведення аналізу існуючих платформ для проведення аукціонів.
2. Визначення архітектури та структури системи.
3. Обґрунтування та вибір технологій для реалізації платформи.
4. Проєктування алгоритмів роботи системи.
5. Реалізація веборієнтованої платформи для проведення аукціонів.
6. Тестування реалізованого програмного продукту.

Об'єктом дослідження є платформа для проведення аукціонів, яка надає можливість користувачам створювати та брати участь в аукціонних торгах, здійснювати ставки, відстежувати хід аукціонів та оформлювати замовлення після завершення торгів.

Предметом дослідження є використання веборієнтованих технологій для створення платформи, яка автоматизує процеси проведення аукціонів, включаючи механізми торгів, забезпечення безпеки та прозорості фінансових транзакцій.

За темою випускної кваліфікаційної роботи бакалавра було опубліковано тези [1,2].

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ОРГАНІЗАЦІЇ ТА УПРАВЛІННЯ ПЛАТФОРМОЮ ДЛЯ ПРОВЕДЕННЯ АУКЦІОНІВ

1.1 Постановка задачі

Метою випускної кваліфікаційної роботи бакалавра є розробка платформи для проведення аукціонів, що забезпечує прозорість, безпеку та ефективність торгів. Така платформа має створити зручний і надійний процес купівлі й продажу, надати користувачам унікальний досвід, сприяти залученню широкої аудиторії та популяризувати рідкісні й цінні речі.

Система онлайн-аукціонів повинна складатися з кількох взаємопов'язаних програмних модулів, кожен з яких відповідає за окрему функціональність платформи:

1. Модуль управління лотами забезпечує можливість продавцям створювати, редагувати та публікувати лоти, а учасникам аукціону переглядати інформацію про лоти.
2. Модуль системи ставок забезпечує можливість розміщення ставок у реальному часі, включаючи функціонал автоматичних ставок, що підвищує зручність участі в торгах.
3. Модуль реєстрація користувачів забезпечує можливість створення особистих кабінетів, де зберігається історія участі в аукціонах, активні лоти та персональні налаштування.
4. Платіжний модуль забезпечує зручну та безпечну обробку транзакцій, дотримуючись сучасних стандартів фінансової безпеки.
5. Модуль адміністрування забезпечує управління користувачами, модерацію лотів та аукціонів, аналіз статистики аукціону та налаштування ключових параметрів платформи.

У результаті повинна бути створена функціональна платформа для проведення аукціонів, на якій будь-хто охочий зможе виставити свій лот на торги або взяти участь у торгах.

Основними етапами розробки платформи є:

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						9

1. Розробка дизайну вебсайту:
 - створення концепції дизайну;
 - розробка UI-компонентів;
 - дизайн вебсторінок.
2. Побудова архітектури клієнтської частини додатку.
 - вибір технологій для реалізації Front End;
 - визначення структури компонентів і сторінок;
 - реалізація управління станом;
 - інтеграція з Back End;
 - адаптивність і підтримка різних пристроїв.
3. Побудова архітектури серверної частини додатку.
 - вибір технологій для реалізації Back End;
 - проектування структури БД;
 - розробка API;
 - авторизація та автентифікація;
 - реалізація модулів для роботи з лотами, аукціонами, користувачами;
 - підтримка WebSocket для ставок у реальному часі;
 - захист API та валідація запитів.
4. Розробка основного функціоналу:
 - розробка логіки реєстрації та авторизації користувачів;
 - розробка логіки створення аукціонів та лотів;
 - реалізація логіки проведення аукціонів;
 - інтеграція платіжної системи;
 - розробка адміністративної частини.
5. Тестування та оптимізація:
 - ручне та автоматизоване тестування програмного продукту;
 - оптимізація роботи програмного продукту.

Результатом виконання поставленого завдання є платформа для проведення аукціонів, що надає користувачам зручні інструменти для участі в

торгах, забезпечує прозорість взаємодії та сприяє популяризації широкого спектра товарів.

1.2 Аналіз існуючих рішень

Сучасний ринок аукціонів пропонує безліч платформ для організації онлайн аукціонів, кожна з яких має свої особливості. У даному розділі розглянуто три відомі платформи для аукціонів – Surplex, Prozorro.Sale та eBay, а також проаналізовано їхні переваги та недоліки.

Surplex – це платформа для проведення аукціонів, що спеціалізується на продажу промислового обладнання, техніки, а також машин і устаткування для підприємств (рис. 1.1) [3]. Нижче наведено її переваги та недоліки.

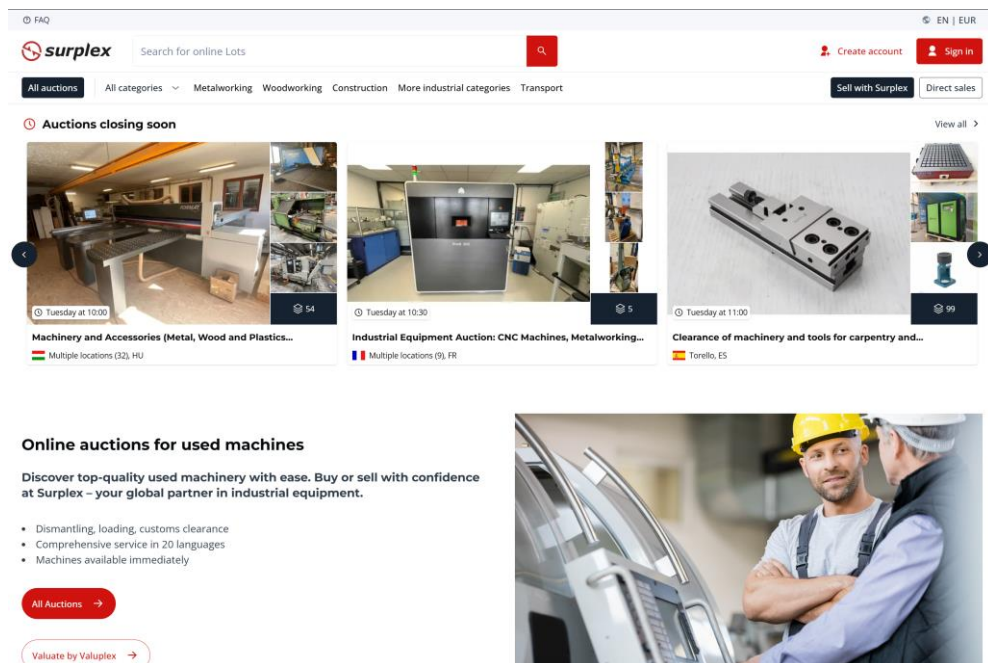


Рис. 1.1 – Платформа для проведення аукціонів Surplex

Основними перевагами Surplex є:

- інтуїтивно зрозумілий інтерфейс;
- розширена фільтрація та пошук лотів;
- детальна інформація про лоти;
- наявність мобільного застосунку.

Основними недоліками Surplex є:

- відсутність провайдерів авторизації;
- можлива неактуальність курсу валют.

Prozorro.Sale – державна електронна аукціонна платформа для продажу та оренди державного та комунального майна в Україні (рис. 1.2) [4]. Нижче наведено її переваги та недоліки.

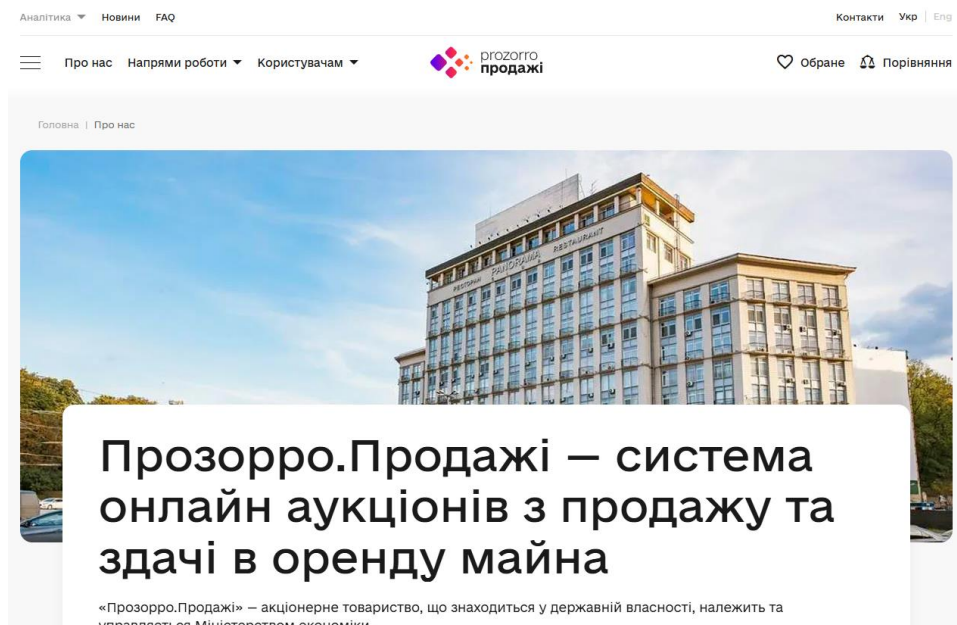


Рис. 1.2 – Платформа для проведення аукціонів Prozorro.Sale

Основними перевагами Prozorro.Sale є:

- публічний API;
- відкритість коду;
- легкість доступу та реєстрації;
- сучасний дизайн;
- інтуїтивно зрозумілий інтерфейс;
- публічні звіти та статистика.

Основними недоліками Prozorro.Sale є:

- відсутність провайдерів авторизації;
- помилки в html які впливають на SEO оптимізацію.

eBay – глобальна платформа для аукціонів і продажів, що пропонує товари різних категорій із можливістю вибору між аукціонами та фіксованими цінами (рис. 1.3) [5]. Забезпечує безпечні транзакції через інтеграцію з PayPal, а також широкий вибір лотів для покупців. Нижче наведено її переваги та недоліки.

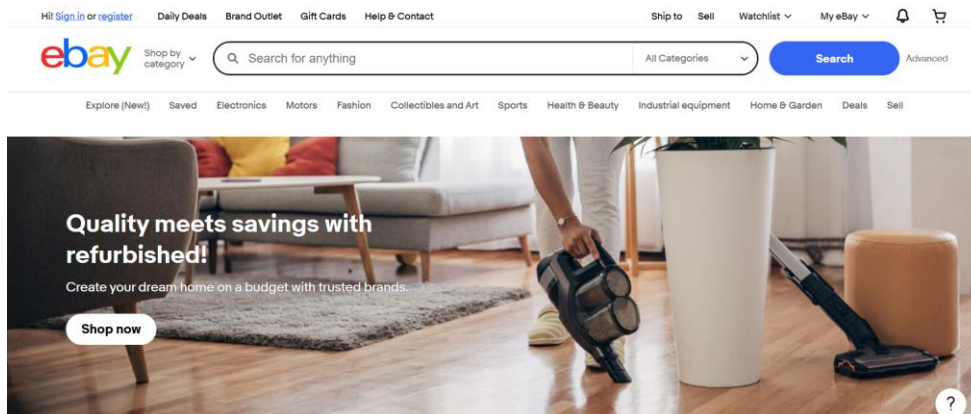


Рис. 1.3 – Платформа для проведення аукціонів eBay

Основними перевагами eBay є:

- публічний API;
- глобальна аудиторія;
- різноманітність лотів;
- інтеграція з платіжними системами;
- наявність мобільного застосунку;
- гнучкість продажів (вибір між аукціоном і фіксованою ціною).

Основними недоліками eBay є:

- застарілий дизайн сайту;
- проблеми з мобільною адаптацією.

Таблиця 1.1

Таблиця порівняння функціональних можливостей платформ

Характеристика	Surplex	Prozorro.Sale	eBay	Моя платформа
Підтримка багатомовності	+	+	+	+
Зручність користування інтерфейсом	+	+	+	+
Різнорманітність лотів	-	+-	+	+
Сучасний дизайн сайту	+	+	-	+
Наявність провайдерів авторизації	-	-	+	+

Інтеграція з платіжними системами	+	+	+	+
-----------------------------------	---	---	---	---

Проведений аналіз демонструє, що на ринку вже існують потужні аукціонні платформи з різноманітними функціональними рішеннями. Це відкриває можливість для створення нової платформи, яка, спираючись на досвід конкурентів, має змогу запропонувати більш цілісне та сучасне рішення для проведення онлайн-аукціонів.

1.3 Вибір архітектури

Для розробки вебзастосунку обрано клієнт-серверну архітектуру (рис 1.4), яка ефективно розподіляє завдання між фронтенд і бекенд частинами системи.

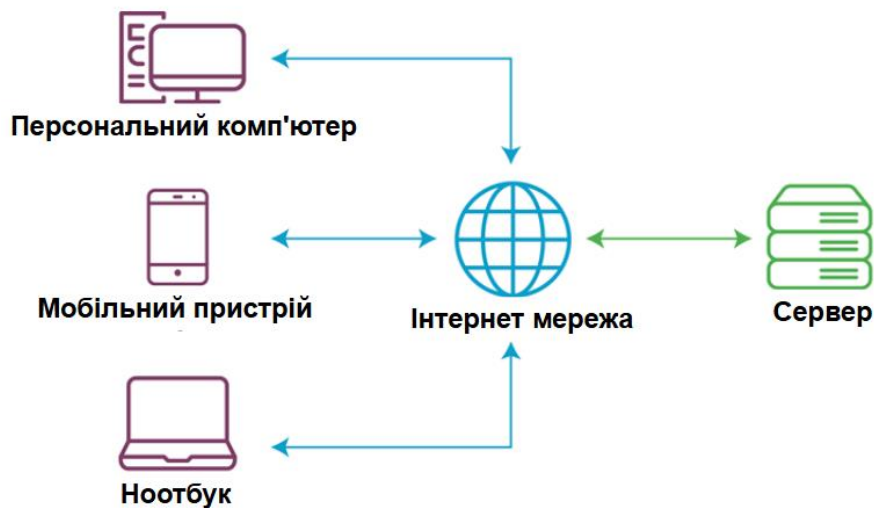


Рис. 1.4 – Клієнт-серверна архітектура

Клієнтська частина забезпечує інтуїтивно зрозумілий та зручний інтерфейс для користувачів, які беруть участь в аукціонах. Реалізація клієнтської частини виконується за допомогою React у фреймворку Next.js, який поєднує можливості серверного рендерингу (SSR) та клієнтського рендеренгу (CSR). Такий підхід дозволяє досягти високої швидкості завантаження сторінок, забезпечуючи SEO-оптимізацію та можливість динамічного оновлення даних на клієнтській стороні.

Навігація між сторінками реалізована за допомогою вбудованої маршрутизації Next.js. Завдяки цьому вебплатформа підтримує клієнтське

оновлення сторінок без необхідності повного перезавантаження, що підвищує зручність для користувача платформи.

Серверна частина платформи відповідає за реалізацію бізнес-логіки, обробку користувацьких запитів, керування базою даних та підтримку масштабованості. Вона реалізована як частина екосистеми Next.js, що дозволяє використовувати API-запити для роботи з даними. Сервер підтримує REST API (рис 1.5), забезпечуючи обробку запитів типу GET, POST, PUT та DELETE для управління даними.

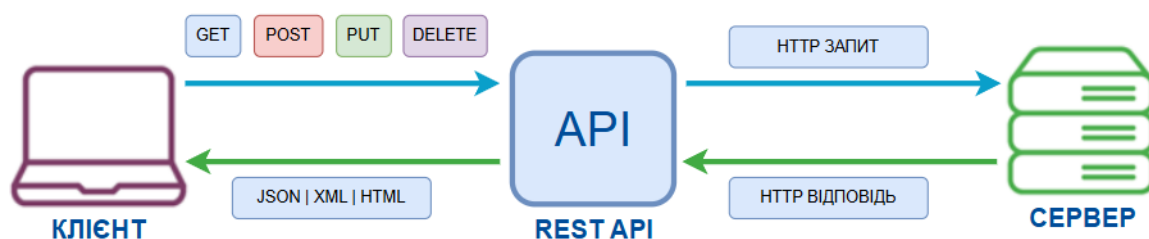


Рис. 1.5 – REST API архітектура

Для роботи з базою даних, WebSocket-з'єднанням і AI-моделями використовується паттерн “Одинак” [6], що забезпечує ініціалізацію єдиного екземпляра кожного з компонентів у межах додатка. Такий підхід мінімізує витрати ресурсів і дозволяє ефективно управляти доступом до цих інстансів.

1.4 Обґрунтування вибору інструментальних засобів

Для розробки платформи проведення аукціонів було обрано сучасний набір технологій, які забезпечують ефективну взаємодію між клієнтською та серверною частинами, зручність у розробці, високу продуктивність та гнучкість масштабування.

Клієнтська частина:

- мова програмування: TypeScript [7];
- фреймворк: Next.js [8];
- бібліотека для інтерфейсу: React [9];
- UI-бібліотеки: Radix UI, ShadCN [10], Tailwind CSS [11];
- бібліотека для роботи з запитами: React Query [12];
- сервіс Cloudinary [13] для зберігання медіаконтенту.

React є провідною бібліотекою для створення динамічних інтерфейсів користувача, що ідеально підходить для реалізації проєкту, оскільки дозволяє розробляти багаторазові компоненти та ефективно управляти станом додатку. Використання Next.js як фреймворку забезпечує додаткові переваги, зокрема серверний рендеринг (SSR), клієнтський рендеринг (CSR) та оптимізовану маршрутизацію, що критично важливо для швидкого відображення аукціонів та взаємодії користувачів із платформою в реальному часі.

TypeScript підвищує надійність коду завдяки статичній типізації, що особливо важливо при роботі з фінансовими операціями та процесами аукціонів.

Для створення сучасного, адаптивного дизайну використовуються Radix UI, ShadCN і Tailwind CSS, які пришвидшують процес розробки та забезпечують гарний вигляд інтерфейсу платформи. React Query оптимізує керування асинхронними запитами, кешуванням і оновленням даних аукціонів.

Серверна частина

- мова програмування: TypeScript;
- фреймворк: Next.js (API Routes);
- аутентифікація: NextAuth.js [14];
- база даних: MongoDB [15];
- ORM: Prisma [16].

Next.js API Routes дозволяють реалізувати серверну логіку без необхідності створення окремого бекенд-сервера. Це спрощує архітектуру проєкту та дозволяє зосередитись на бізнес-логіці аукціонів.

Для реалізації аутентифікації обрано NextAuth.js – готове рішення, що підтримує різні методи авторизації, зокрема через email/пароль та популярні OAuth-провайдери (Google, GitHub тощо), забезпечуючи надійну систему автентифікації користувачів платформи.

Prisma як ORM забезпечує типобезпечну взаємодію з базою даних, що зменшує кількість помилок при роботі з даними.

Після аналізу різних систем управління базами даних було обрано MongoDB за наступними критеріями (табл. 1.2):

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						16

Аналіз показників

Критерій	PostgreSQL	MongoDB	MySQL
Тип схеми	Статична	Динамічна	Статична
Масштабованість	Горизонтальна та вертикальна	Горизонтальна, але складніша	Вертикальна
Продуктивність на запис	Висока	Висока, але залежить від структури даних	Висока
Гнучкість запитів	Висока (SQL)	Висока (гнучкий JSON)	Висока (SQL)
Взаємозв'язки	Чіткі реляційні зв'язки	Посилання або вбудовані документи	Чіткі реляційні зв'язки
Вбудоване кешування	Так	Так	Так
Підтримка великої кількості записів	Обмежена через реляційну структуру	Оптимізована для великих обсягів даних	Обмежена
Гнучкість структури даних	Низька (строга схема)	Висока (схема необов'язкова)	Низька (строга схема)

MongoDB було обрано для платформи аукціонів завдяки її гнучкості, масштабованості та продуктивності. Відсутність жорсткої схеми дозволяє змінювати структуру даних без міграцій. Використання зв'язків між колекціями дає можливість ефективно працювати із взаємопов'язаними даними. Горизонтальне масштабування забезпечує стабільну роботу при зростанні навантаження.

У процесі розробки використовував різні інтегровані середовища розробки та програмні інструменти, що забезпечували ефективність роботи. Серед них – Visual Studio Code для написання коду, Studio 3T для роботи з базою даних, а також графічні редактори draw.io для створення схем і діаграм та Adobe Photoshop для обробки графічних матеріалів.

Висновок до першого розділу

У першому розділі проведено аналіз вимог до платформи для проведення аукціонів, визначено ключові компоненти та етапи її розробки. Досліджено аналоги, зокрема Surplex, Prozorro.Sale та eBay, що дозволило виявити їхні переваги, недоліки та визначити важливі аспекти для власної платформи.

На основі аналізу обрано клієнт-серверну архітектуру, яка забезпечує ефективний розподіл обчислювального навантаження та масштабованість системи.

Для реалізації клієнтської частини вирішено використовувати React у фреймворку Next.js, що поєднує можливості серверного та статичного рендерингу, підвищуючи продуктивність і SEO-оптимізацію. Серверна частина побудована на REST API, що забезпечує зручну взаємодію з базою даних та іншими сервісами.

Таким чином, у цьому розділі було сформовано загальну концепцію платформи, визначено технологічний стек і закладено основу для подальшої реалізації системи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАТФОРМИ ДЛЯ ПРОВЕДЕННЯ АУКЦІОНІВ

2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

Аукціонна платформа надає інтерфейс, за допомогою якого користувачі можуть створювати власні аукціони або брати участь в аукціонах інших користувачів. В системі передбачено дві ролі: користувач та адміністратор. Користувач має можливість як виставляти на торги власні товари, створюючи аукціони, так і брати участь у торгах інших користувачів з метою виграшу лотів. Такий підхід забезпечує гнучку модель участі в аукціонах.

Кожен лот містить опис товару, стартову ціну, крок ставки, тривалість аукціону та іншу важливу інформацію. Адміністратор системи здійснює контроль за змістом платформи, модерує контент та має доступ до управління користувачами.

Для наочного представлення взаємодії між ролями та компонентами системи було створено діаграму варіантів використання (рис. 2.1) [17], яка ілюструє основні сценарії роботи платформи. В аукціонній платформі розрізняють два ключових актори: адміністратор (admin) та користувач (user).

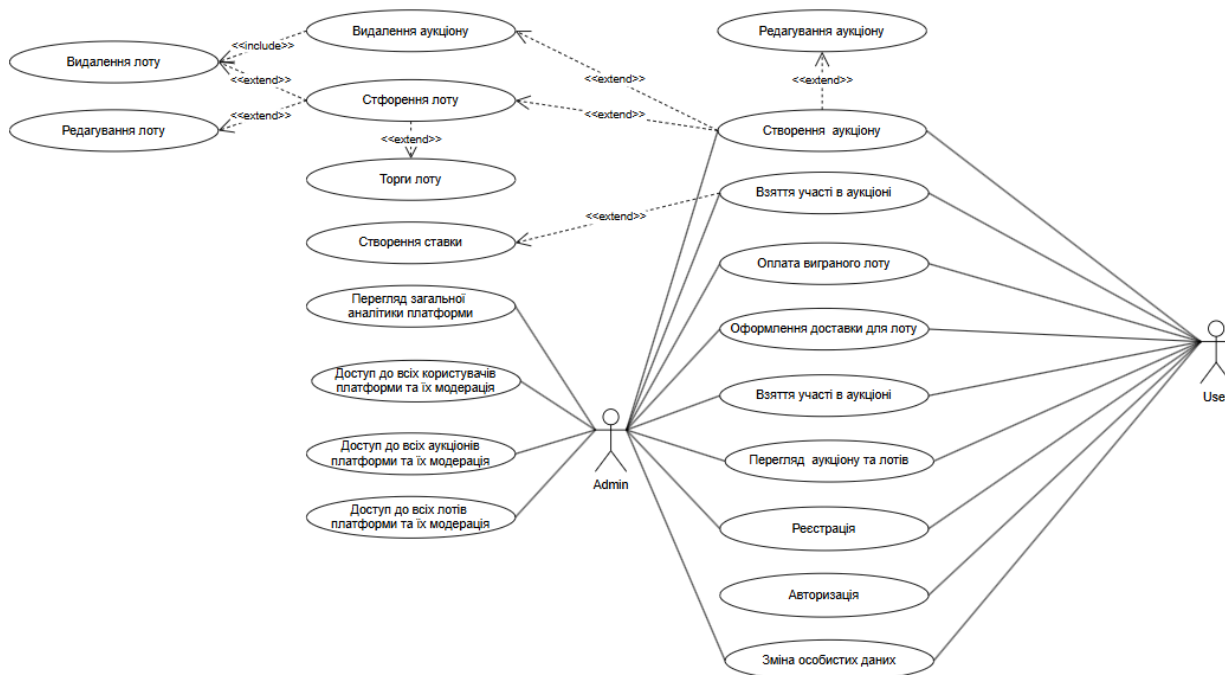


Рис. 2.1 – Діаграма варіантів використання системи

Короткий опис акторів представлено в табл. 2.1.

Таблиця 2.1

Виявлення акторів

Актор	Короткий опис
Адміністратор (Admin)	Здійснює модерацію користувачів і контенту. Має доступ до усіх частин сайту
Користувач (User)	Може переглядати лоти, робити ставки, створювати аукціони та лоти для цих аукціонів

У процесі розробки аукціонної платформи було визначено основні варіанти використання, які описують взаємодію користувачів із системою. Вони охоплюють базові функції, необхідні для забезпечення зручної роботи з платформою, як для звичайних користувачів, так і для адміністраторів.

У таблиці 2.2 представлено основні варіанти використання, що відображають ключові дії, які можуть виконувати користувачі та адміністратори в системі.

Таблиця 2.2

Виявлення варіантів використання

Основний актор	Найменування	Формулювання
Користувач/ Адміністратор	Авторизація	Цей варіант використання дозволяє користувачеві ввести свої облікові дані (логін та пароль) для отримання доступу до персоналізованих функцій платформи або здійснити авторизацію за допомогою провайдерів (Google, Github).
	Реєстрація	Цей варіант використання дозволяє новому користувачеві створити обліковий запис у системі, надавши необхідні особисті дані для подальшого доступу до платформи.
	Зміна особистих даних	Цей варіант використання дозволяє користувачеві редагувати свою особисту інформацію (пароль)
	Перегляд аукціонів та лотів	Цей варіант використання дозволяє користувачеві ознайомлюватися з аукціонами, переглядати їх опис, а також детальну інформацію про лоти.

Продовження таблиці 2.2

	Взяття участі в аукціоні	Цей варіант використання дозволяє користувачеві зробити ставки на лоти під час активності аукціону та змагатися за право покупки
	Оформлення доставки для лоту	Цей варіант використання дозволяє переможцю аукціону вказати адресу доставки
	Оплата виграного лоту	Цей варіант використання дає змогу здійснити оплату за лот, який користувач виграв в аукціоні.
	Створення аукціону	Цей варіант використання дозволяє користувачеві створити новий аукціон, вказавши його назву, опис, тривалість та інші параметри
	Редагування аукціону	Цей варіант використання дає змогу змінювати інформацію про вже створений аукціон
	Створення ставки	Цей варіант використання дозволяє користувачеві зробити ставку для лоту, вказавши суму ставки та підтвердивши її через інтерфейс платформи
	Створення лоту	Цей варіант використання дозволяє користувачеві створити новий лот в аукціоні, вказавши його заголовок, опис, мінімальну ставку та інші параметри
	Видалення лоту	Цей варіант використання дозволяє користувачеві видалити лот з аукціону, що припиняє його участь у торгах і прибирає його з доступу на платформі
	Редагування лоту	Цей варіант використання дозволяє користувачеві змінити деталі лоту, такі як опис, заголовок, мінімальна ставка та інші параметри
	Торги лоту	Цей варіант використання дозволяє користувачам робити ставки на лот, підвищуючи його ціну під час активних торгів, що сприяє конкуренції між учасниками
	Видалення аукціону	Цей варіант використання дозволяє користувачеві видалити аукціон

Адміністратор	Перегляд загальної аналітики платформи	Цей варіант використання дозволяє адміністратору переглядати загальну аналітику платформи, зокрема статистика приросту аукціонів, лотів та користувачів, що допомагає відслідковувати ефективність роботи платформи
	Доступ до всіх користувачів платформи та їх модерація	Цей варіант використання дозволяє адміністратору отримати доступ до інформації про всіх зареєстрованих користувачів платформи, а також модерацію їх акаунтів, включаючи можливість деактивацію користувача.
	Доступ до всіх аукціонів платформи та їх модерація	Цей варіант використання дозволяє адміністратору переглядати всі активні та завершені аукціони на платформі, а також здійснювати модерацію аукціонів, включаючи їх редагування або видалення за необхідності
	Доступ до всіх лотів платформи та їх модерація	Цей варіант використання дозволяє адміністратору отримати доступ до всіх лотів на платформі, переглядати інформацію, редагувати деталі або видаляти за необхідності

Під час проєктування платформи було визначено основні функціональні та нефункціональні вимоги, що забезпечують коректну роботу системи, зручність для користувачів та відповідність технічним критеріям.

Функціональні вимоги

1. Система повинна забезпечувати реєстрацію, аутентифікацію та авторизацію користувачів, з підтримкою різних ролей (користувач, адміністратор).
2. Система повинна дозволяти користувачам створювати, редагувати, видаляти та публікувати аукціони.
3. Система повинна надавати можливість додавати опис, фото та відео до лотів.
4. Система повинна інтегруватися з платіжними системами для здійснення оплати та забезпечувати захист фінансових транзакцій.

5. Система повинна забезпечувати можливість здійснення ставок у реальному часі, з автоматичним оновленням інформації про поточні ставки.
6. Система повинна дозволяти адміністраторам змінювати статуси лотів.
7. Система повинна надавати користувачам доступ до історії ставок та результатів аукціонів.
8. Система повинна дозволяти адміністраторам здійснювати модерацію лотів і користувачів.
9. Система повинна дозволяти користувачам оформлювати замовлення на придбання лотів.

Нефункціональні вимоги

1. Часова ефективність:
 - час реакції програми на дії користувача (введення даних, ставки, перегляд лотів) не повинен перевищувати 1 секунди;
 - оновлення ставок у реальному часі повинно відбуватися з затримкою не більше 2 секунд;
 - відображення масивів даних (активні лоти, історія ставок, результати аукціонів) за запитами користувачів не більше 5 секунд.
2. Доступність системи:
 - доступність бази даних повинна становити 99%;
 - операції підключення до БД не повинні перевищувати 2 секунди.
3. Пам'ять та ресурси:
 - система повинна забезпечувати стабільну роботу з мультимедійними файлами, зберігаючи їх у Cloudinary та підтримуючи високу швидкість завантаження навіть при великій кількості запитів;
 - обсяг оперативної пам'яті, необхідний для стабільної роботи програми, має бути не менше 4 Гб.
4. Надійність та доступність:
 - система повинна бути адаптована для використання на мобільних пристроях з оптимізованим інтерфейсом;

- сайт повинен стабільно функціонувати за умови безперебійної роботи серверного обладнання та доступу до мережі Інтернет;
- доступність сайту має становити 99.9% навіть при одночасному доступі до 5000 користувачів;
- інтерфейс сайту має коректно відображатися та функціонувати в актуальних версіях основних браузерів (Google Chrome, Mozilla Firefox, Safari).

2.2 Розробка бази даних системи

Дані в системі зберігаються в базі даних MongoDB, яка складається з 12 таблиць (табл. 2.3).

Таблиця 2.3

Таблиці БД

Назва таблиці	Призначення
User	Таблиця користувачів
Account	Таблиця акаунтів користувачів
Auction	Таблиця аукціонів
UserAuction	Таблиця зв'язку користувачів з аукціоном
Lot	Таблиця лотів
Bid	Таблиця ставок у лотах
Category	Таблиця категорій лотів
LotCategory	Таблиця зв'язку лотів з категоріями
LotDetail	Таблиця деталей лотів
Photo	Таблиця фотографій лотів
Video	Таблиця відео лотів
Shipping	Таблиця для збереження інформації про доставку

Структура бази даних зображена на (рис 2.2).

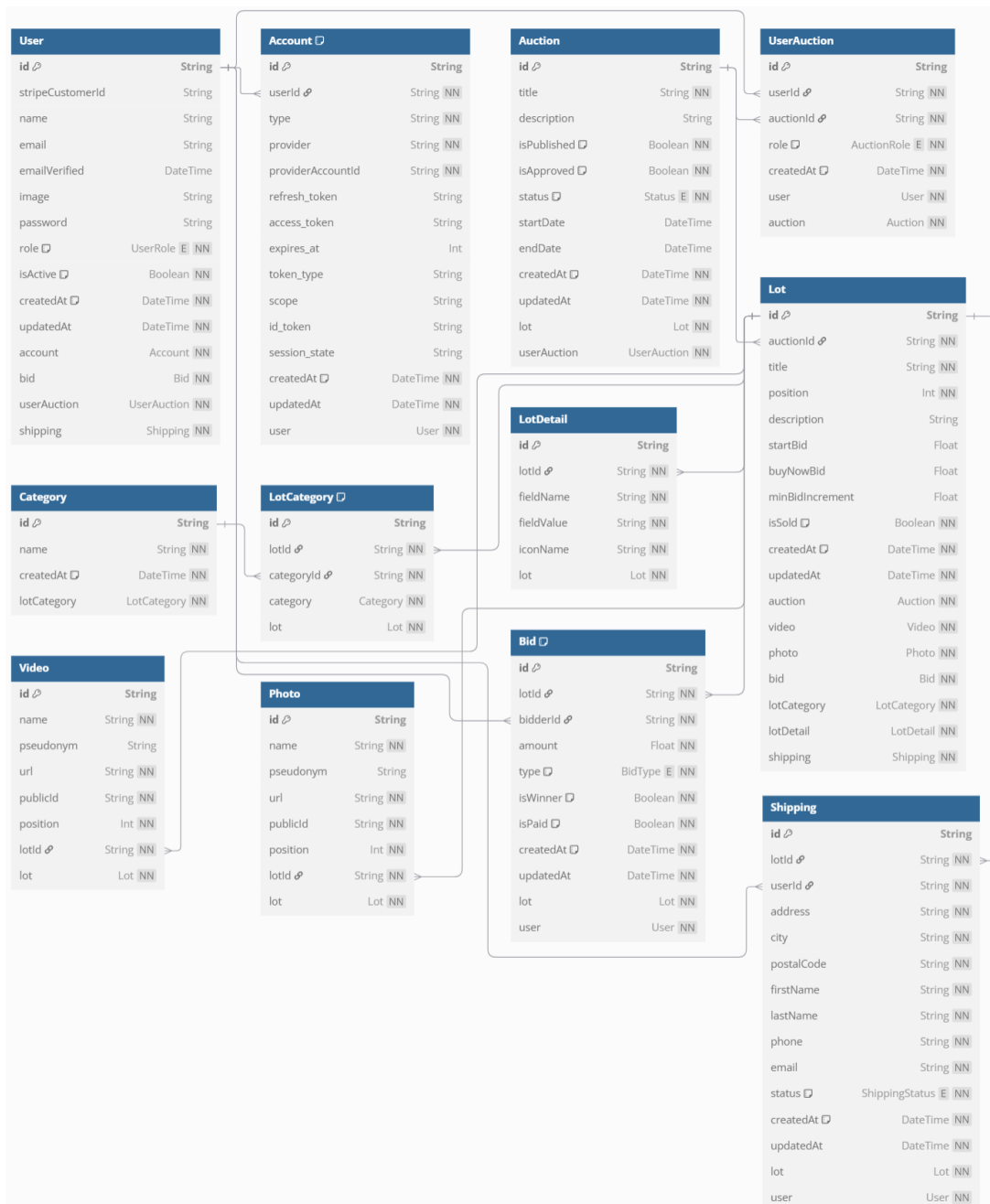


Рис 2.2 – Структура БД

Таблиця “User” слугує для збереження даних користувача. Опис полів таблиці наведений нижче (табл. 2.4).

Таблиця 2.4

Структура таблиці “User”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
name	String	-	-	Ім'я
email	String	-	-	Електронна пошта
emailVerified	DateTime	-	-	Час підтвердження пошти

Продовження таблиці 2.4

stripeCustomerId	String	-	-	Ідентифікатор користувача в Stripe
image	String	-	-	Посилання на фото профілю
password	String	-	-	Пароль
role	String	-	-	Роль
isActive	Boolean	-	-	Визначає доступ до системи
createdAt	DateTime	-	-	Дата створення облікового запису
updatedAt	DateTime	-	-	Дата останнього оновлення

Таблиця “Account” слугує для збереження інформацію про способи автентифікації користувача. Опис полів таблиці наведений нижче (табл. 2.5).

Таблиця 2.5

Структура таблиці “Account”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор облікового запису
userId	String	-	User.id	Ідентифікатор користувача
type	String	-	-	Тип облікового запису
provider	String	-	-	Провайдер автентифікації
providerAccountId	String	-	-	Ідентифікатор у провайдера
refresh_token	String	-	-	Токен оновлення
access_token	String	-	-	Токен доступу
expires_at	Int	-	-	Час закінчення дії токена
token_type	String	-	-	Тип токена
scope	String	-	-	Доступні області застосування токена
id_token	String	-	-	Ідентифікаційний токен
session_state	String	-	-	Стан сесії
createdAt	DateTime	-	-	Дата створення
updatedAt	DateTime	-	-	Дата оновлення

Таблиця “Auction” слугує для збереження інформацію про аукціон. Опис полів таблиці наведений нижче (табл. 2.6).

Таблиця 2.6

Структура таблиці “Auction”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
title	String	-	-	Назва
description	String	-	-	Опис
isPublished	Boolean	-	-	Ознака публікації
isApproved	Boolean	-	-	Ознака успішного підтвердження лота
status	String	-	-	Статус

startDate	DateTime	-	-	Дата початку
endDate	DateTime	-	-	Дата завершення
createdAt	DateTime	-	-	Дата створення
updatedAt	DateTime	-	-	Дата оновлення

Таблиця “Lot” містить інформацію про лоти аукціону. Опис полів таблиці наведений нижче (табл. 2.7).

Таблиця 2.7

Структура таблиці “Lot”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
auctionId	String	-	Auction.id	Ідентифікатор аукціону
title	String	-	-	Назва
position	Int	-	-	Позиція
description	String	-	-	Опис
startBid	Float	-	-	Початкова ставка
buyNowBid	Float	-	-	Ціна "Купити зараз"
minBidIncrement	Float	-	-	Мінімальне підвищення ставки
isSold	Boolean	-	-	Ознака проданого лота
createdAt	DateTime	-	-	Дата створення
updatedAt	DateTime	-	-	Дата оновлення

Таблиця “Bid” містить інформацію про ставки, зроблені на лоти. Опис полів таблиці наведений нижче (табл. 2.8).

Таблиця 2.8

Структура таблиці “Bid”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
lotId	String	-	Lot.id	Ідентифікатор лота
bidderId	String	-	User.id	Ідентифікатор користувача, який зробив ставку
amount	Float	-	-	Розмір
type	String			Тип ставки
isWinner	Boolean	-	-	Флаг для визначення чи є ставка переможною
isPaid	Boolean	-	-	Флаг для визначення чи є ставка оплаченою
createdAt	DateTime	-	-	Дата та час створення
updatedAt	DateTime	-	-	Дата та час оновлення

Таблиця “Photo” містить інформацію про фотографії лотів. Опис полів таблиці наведений нижче (табл. 2.9).

Структура таблиці “Photo”

Назва	Тип даних	PK	FK	Опис поля
id	ObjectId	+	-	Унікальний ідентифікатор
name	String	-	-	Назва
pseudonym	String	-	-	Псевдонім
url	String	-	-	Посилання
publicId	String	-	-	Ідентифікатор фото на платформі Cloudinary
position	Int	-	-	Порядок відображення фото в галереї
lotId	String	-	Lot.id	Ідентифікатор лота, до якого належить фото

Таблиця “Video” містить інформацію про відео лотів. Опис полів таблиці наведений нижче (табл. 2.10).

Таблиця 2.10

Структура таблиці “Video”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
name	String	-	-	Назва
pseudonym	String	-	-	Псевдонім
url	String	-	-	Посилання
publicId	String	-	-	Ідентифікатор відео на платформі Cloudinary
position	Int	-	-	Порядок відображення відео в галереї
lotId	String	-	Lot.id	Ідентифікатор лота, до якого належить відео

Таблиця “LotCategory” пов’язує лоти з категоріями. Опис полів таблиці наведений нижче (табл. 2.11).

Таблиця 2.11

Структура таблиці “LotCategory”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
lotId	String	-	Lot.id	Ідентифікатор лота
categoryId	String	-	Category.id	Ідентифікатор категорії

Таблиця “LotDetail” містить детальну інформацію про деталі лоту. Опис полів таблиці наведений нижче (табл. 2.12).

Таблиця 2.12

Структура таблиці “LotDetail”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
lotId	String	-	Lot.id	Ідентифікатор лота
fieldName	String	-	-	Назва характеристики
fieldValue	String	-	-	Значення характеристики
iconName	String	-	-	Назва іконки

Таблиця “Category” містить перелік категорій лотів. Опис полів таблиці наведений нижче (табл. 2.13).

Таблиця 2.13

Структура таблиці “Category”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
name	String	-	-	Назва категорії
createdAt	DateTime	-	-	Дата створення запису

Таблиця “UserAuction” відображає зв’язок між користувачами та аукціонами. Опис полів таблиці наведений нижче (табл. 2.14).

Таблиця 2.14

Структура таблиці “UserAuction”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
userId	String	-	User.id	Ідентифікатор користувача
auctionId	String	-	Auction.id	Ідентифікатор аукціону
role	String	-	-	Флаг який визначає роль користувача
createdAt	DateTime	-	-	Дата створення запису

Таблиця “Shipping” містить інформацію про доставку. Опис полів таблиці наведений нижче (табл. 2.15).

Таблиця 2.15

Структура таблиці “Shipping”

Назва	Тип даних	PK	FK	Опис поля
id	String	+	-	Унікальний ідентифікатор
lotId	String	-	+	Ідентифікатор лота
userId	String	-	+	Ідентифікатор користувача
address	String	-	-	Адреса доставки

city	String	-	-	Місто доставки
postalCode	String	-	-	Індекс доставки
firstName	String	-	-	Ім'я отримувача
lastName	String	-	-	Прізвище отримувача
phone	String	-	-	Телефон
email	String	-	-	Електронна пошта
status	String	-	-	Статус доставки
createdAt	DateTime	-	-	Дата створення запису
updatedAt	DateTime	-	-	Дата оновлення запису

2.3 Проєктування та реалізація алгоритмів роботи системи

Алгоритми програмної системи аукціонної платформи визначають логіку виконання задач, взаємодію між модулями, а також послідовність дій користувача. Чітке опрацювання гарантує стабільність роботи всієї системи, забезпечує передбачуваність поведінки платформи, спрощує впровадження нового функціоналу, а також полегшує процес тестування та усунення помилок.

Серед ключових елементів логіки – процеси автентифікації, авторизації та реєстрації користувачів. Вони є критично важливими для забезпечення безпеки особистих даних, захисту від несанкціонованого доступу, а також створення індивідуального користувацького досвіду в межах платформи.

Діаграма активності на рис. 2.3 ілюструє загальну послідовність дій користувача під час взаємодії з платформою – від ініціації входу до отримання доступу до платформи [18]. Вона дозволяє наочно простежити можливі сценарії: використання наявного облікового запису, проходження автентифікації, або створення нового профілю.

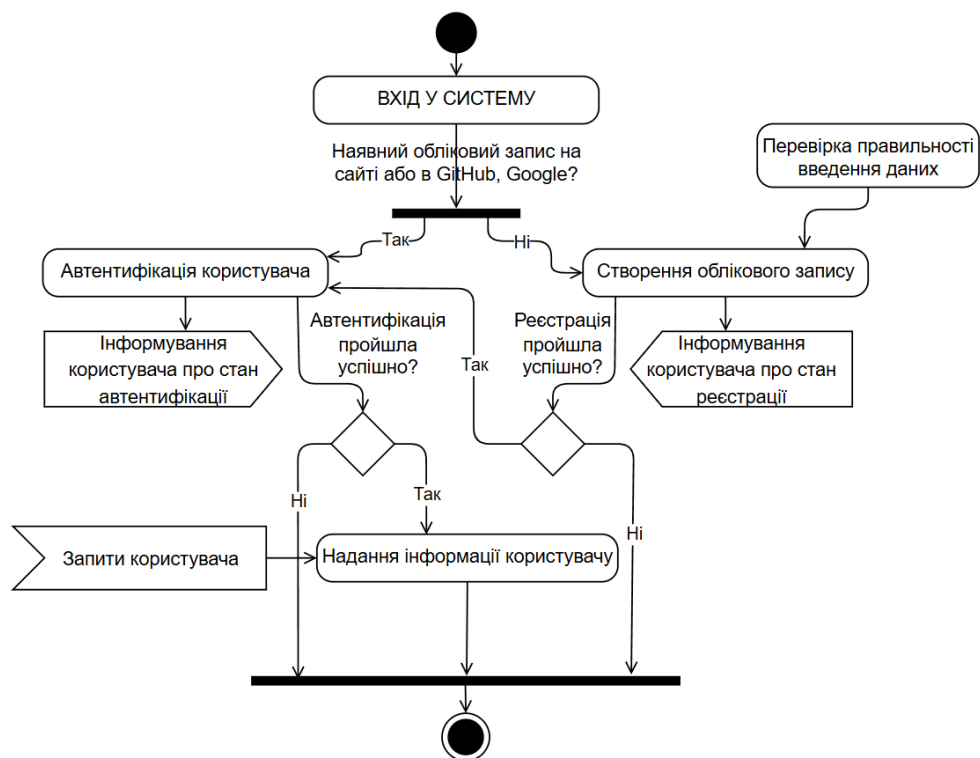


Рис. 2.3 – Діаграма активності входу користувача на платформу

Основні етапи процесу:

1. Ініціація входу до системи

Користувач починає взаємодію з платформою шляхом входу до системи. Це може відбуватися як зі сторінки входу, так і через автоматизовану інтеграцію, наприклад, за допомогою OAuth-сервісів (GitHub, Google тощо)

2. Перевірка наявності облікового запису

Система перевіряє, чи існує обліковий запис користувача. У разі його наявності користувач може пройти процедуру автентифікації, використовуючи або внутрішні механізми (логін/пароль), або зовнішні сервіси (OAuth). Такий підхід підвищує зручність і прискорює процес входу.

3. Автентифікація користувача

У разі наявності облікового запису система переходить до етапу автентифікації. Тут перевіряється правильність введених облікових даних. Якщо дані коректні, система виконує авторизацію – надання доступу до ресурсів, дозволених для даного користувача. У разі помилки користувач отримує відповідне повідомлення, що дозволяє швидко виявити та виправити проблему.

4. Процес реєстрації нового користувача

Якщо обліковий запис відсутній, користувач має можливість зареєструватися, створивши новий профіль. Система пропонує заповнити відповідну форму з обов'язковими полями (ім'я, електронна пошта, пароль тощо) і перевіряє правильність введених даних. У разі успішної реєстрації користувач отримує повідомлення про створення облікового запису та можливість авторизуватися в системі.

5. Надання доступу до функціоналу платформи

Після проходження автентифікації та авторизації, користувач отримує доступ до основних можливостей платформи: перегляду активних аукціонів, участі в торгах, створення нових лотів, перегляду історії ставок, керування власними лотами та налаштуванням профілю. Залежно від ролі користувача, йому можуть бути доступні додаткові функції (можливість модерації та перегляду статистичної інформації роботи платформи).

6. Інформування користувача про стан дій

На кожному етапі користувач отримує повідомлення про успішність або помилки – як у випадку помилок під час введення даних, так і у разі успішної реєстрації чи входу. Це підвищує зручність взаємодії з платформою та дозволяє уникнути непорозумінь.

7. Завершення процесу автентифікації/реєстрації

Після надання доступу до функцій платформи процес входу або реєстрації вважається завершеним. Далі користувач може повноцінно взаємодіяти з системою.

Діаграма послідовності на рис. 2.4 відображає процес взаємодії користувача з платформою під час перегляду лота та створення ставки [19]. Вона демонструє загальну логіку дій і дозволяє простежити послідовність обміну повідомленнями між користувачем, системою та базою даних.

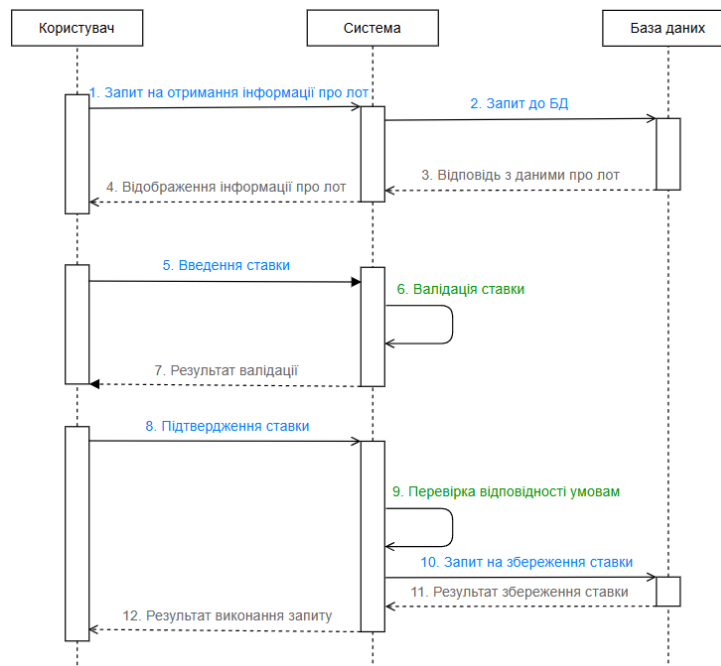


Рис. 2.4 – Діаграма послідовності здійснення ставки для лоту

Процес здійснення ставки для лоту передбачає низку послідовних дій, що забезпечують коректну взаємодію користувача із системою та актуальність інформації про лот. Далі наведено покроковий опис основних етапів.

1. *Користувач надсилає запит на отримання інформації про лот.*

На початковому етапі користувач відкриває сторінку певного лоту з усіх доступних лотів для вибраного аукціону. Система фіксує цю дію як запит на отримання повної інформації про вибраний лот, включаючи його характеристики, історію ставок, статус торгів та іншу інформацію.

2. *Система надсилає відповідний запит до бази даних.*

Щоб сформулювати відповідь для користувача, система звертається до бази даних. Система очікує отримати всю необхідну інформацію для лоту.

3. *База даних повертає інформацію про лот.*

У відповідь на запит система отримує структурований набір даних, що містить усі необхідні поля, пов'язані з лотом. Ці дані передаються далі для обробки.

4. *Система відображає інформацію про лот користувачу.*

Отримані дані перетворюються у візуальний інтерфейс – фото, відео, текстові блоки – й відображаються на екрані. Користувач бачить деталі лоту та отримує можливість прийняти рішення щодо участі в торгах.

5. Користувач вводить ставку на лот.

Після аналізу інформації користувач вирішує зробити ставку. Він вводить суму та підтверджує дію, наприклад, натисканням кнопки "Зробити ставку". Ці дані відправляються на обробку до системи.

6. Система виконує валідацію ставки.

Валідація включає кілька рівнів перевірки: чи є сума числом, чи не є вона меншою за мінімальний крок або поточну максимальну ставку.

7. Результат валідації передається користувачу.

Якщо всі умови дотримані, система надає позитивний зворотний зв'язок. У випадку помилок (наприклад, ставка надто низька або торги завершено) – користувач бачить заблоковану кнопку для здійснення ставки.

8. Користувач підтверджує ставку.

Після вдалого проходження перевірки користувач надсилає кнопку "Здійснити ставку".

9. Система виконує фінальну перевірку відповідності умовам.

Щоб уникнути зміни ситуації під час підтвердження (наприклад, якщо інший користувач зробив ставку в той самий час), система ще раз перевіряє актуальний стан лоту й умови – ставка не повинна бути застарілою чи неактуальною.

10. Система надсилає запит на збереження ставки у базу даних.

Якщо всі перевірки успішні, система формує запит на створення нової ставки у базі даних.

11. База даних обробляє запит і повертає результат.

Запис про нову ставку зберігається у відповідній таблиці. Якщо збереження проходить успішно, запит не викликає помилки, у разі помилки – відповідне повідомлення.

12. Система повідомляє користувача про результат операції.

На фінальному етапі користувач отримує повідомлення про успішне створення ставки. Сторінка оновлюється, і ставка користувача позначається як проведена. У разі проблем відображається повідомлення про помилку.

Діаграма активності на рис. 2.5 демонструє процес участі користувача в аукціоні та автоматизацію системи для визначення переможця. Вона ілюструє взаємодію між двома основними учасниками: користувачем, який здійснює дії (вносить ставки або купує лот), і системою, яка забезпечує перевірку та оновлення інформації про лот. Цей процес охоплює ключові етапи торгів лоту на аукціоні: початок, внесення ставок, миттєву покупку лоту та завершення часу продажу лоту з визначенням переможця.

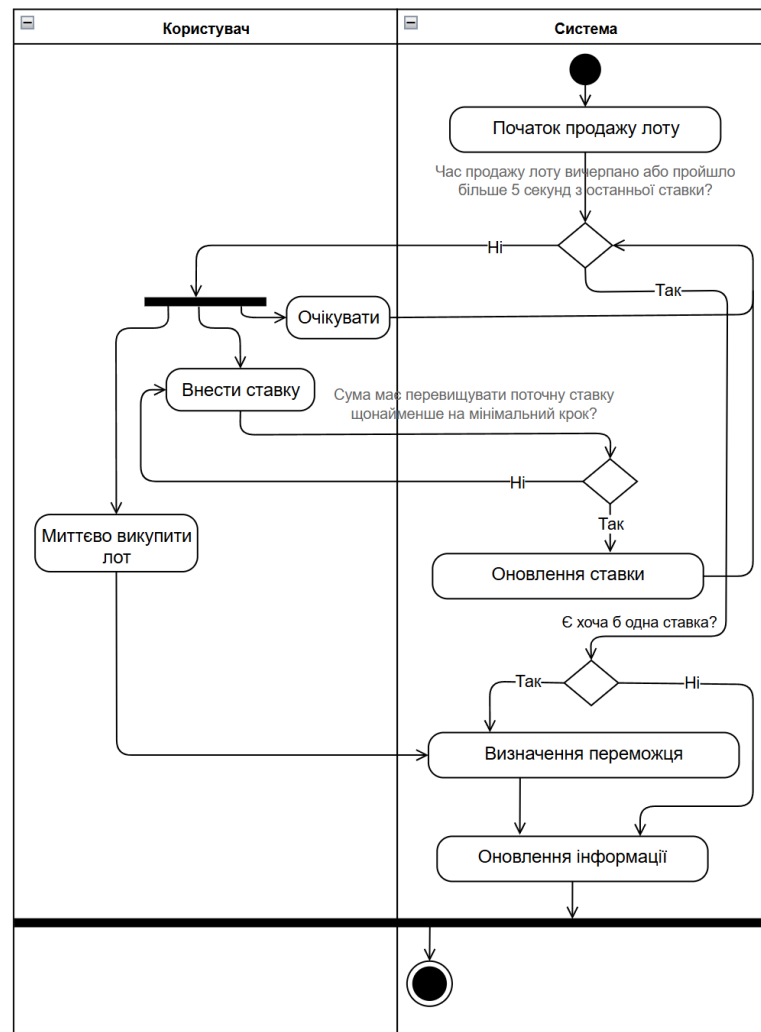


Рис. 2.5 – Діаграма активності процесу торгів на аукціоні

Користувач має два основні способи взаємодії з лотом. Перший – миттєва покупка, яка дозволяє одразу придбати лот без участі в торгах. У цьому випадку система негайно оновлює відповідну інформацію, продаж вважається завершеним, і лот переходить у власність користувача. Другий спосіб – внесення ставки. Якщо користувач бажає взяти участь у торгах, він може зробити ставку, яка обов’язково має перевищувати поточну найвищу ставку.

Зі свого боку система відповідає за ініціювання та контроль процесу продажу лотів. Вона запускає продаж лоту та постійно відслідковує перебіг торгів. Зокрема, система перевіряє, чи не завершився встановлений час продажу, або чи минуло понад 5 секунд із моменту останньої ставки. Якщо час ще не вичерпано, і з моменту останньої ставки минуло не більше 5 секунд, система перевіряє, чи нова ставка є вищою за поточну мінімальну. Якщо так – інформація про лот оновлюється. У разі ж завершення відведеного часу або якщо з моменту останньої ставки минуло понад 5 секунд, система визначає переможця торгів – учасника з найвищою ставкою та відповідно оновлює інформацію.

Після визначення переможця або здійснення миттєвої покупки система остаточно оновлює дані щодо лоту і завершує процес торгів, фіксуючи результат продажу.

Дана діаграма чітко демонструє алгоритм роботи аукціонної платформи, забезпечуючи прозорість дій користувачів і правильність роботи системи.

2.4 Реалізація системи аукціонних торгів

При надходженні запиту на створення аукціону спочатку здійснюється перевірка авторизації користувача. Якщо користувач не авторизований, система повертає відповідь із кодом помилки 401. У випадку, якщо користувач був успішно авторизований, виконується обробка даних, що надійшли у запиті, зокрема перевіряється наявність обов'язкового поля – назви аукціону. Після валідації даних створюється новий запис у таблиці аукціонів бази даних, а також виконується прив'язка створеного аукціону до користувача, який його створив (рис. 2.6).

```

export async function POST(request: Request) {
  try {
    const authUser = await getAuthUser();
    if (!authUser) {
      return new NextResponse('Unauthorized', { status: 401 });
    }

    const body = await request.json();
    const { title } = body;

    if (!title) {
      return new NextResponse('Title is required and cannot be empty', { status: 400 });
    }

    const auction = await prismaDb.auction.create({
      data: {
        title,
      },
    });

    await prismaDb.userAuction.create({
      data: {
        userId: authUser.id,
        auctionId: auction.id,
      },
    });

    return NextResponse.json(auction);
  } catch (error) {
    console.error('CREATE_AUCTION_ERROR -> ', error);
    return new NextResponse('Internal server error', { status: 500 });
  }
}

```

Рис. 2.6 – Алгоритм створення нового аукціону

Алгоритм створення нового лота реалізується через обробку POST-запиту. Спочатку здійснюється перевірка авторизації користувача та валідність переданих даних, таких як назва лота та ідентифікатор аукціону. У разі некоректних даних запит завершується помилкою. Якщо всі перевірки проходять успішно, система шукає відповідний аукціон у базі даних. Логіка роботи продемонстрована на рис. 2.7.

```

export async function POST(request: Request, { params }: { params: { auctionId: string } }) {
  try {
    const authUser = await getAuthUser();
    if (!authUser) {
      return new NextResponse('Unauthorized', { status: 401 });
    }

    const body = await request.json();
    const { title } = body;

    if (!title) {
      return new NextResponse('Title is required and cannot be empty', { status: 400 });
    }

    const auctionId = params.auctionId;
    if (!auctionId) {
      return new NextResponse('Auction is required and cannot be empty', { status: 400 });
    }

    const auction = await prismaDb.auction.findFirst({
      where: {
        id: auctionId,
      },
      include: {
        lot: true,
      },
    });

    if (!auction) {
      return new NextResponse('Auction not found', { status: 404 });
    }

    const lotPosition = auction.lot?.length
      ? auction.lot.reduce((max, lot) => (lot.position > max.position ? lot : max)).position + 1
      : 1;

    const lot = await prismaDb.lot.create({
      data: {
        title,
        auctionId: auction.id,
        position: lotPosition,
      },
    });

    return NextResponse.json(lot);
  } catch (error) {
    console.error('CREATE_LOT_ERROR -> ', error);
    return new NextResponse('Internal server error', { status: 500 });
  }
}

```

Рис. 2.7 – Алгоритм створення нового лоту

Після знаходження аукціону визначається позиція нового лота: якщо існують інші лоти, вона обчислюється як максимальна позиція плюс один; якщо лотів ще немає – позиція дорівнює одиниці. Створений лот зберігається в базі даних із вказаними параметрами, після чого формується відповідь із інформацією про новий запис.

Алгоритм обробки ставок через WebSocket-з'єднання реалізується шляхом ініціалізації сервера Socket.io на стороні API-обробника. Після створення інстансу сервера відбувається налаштування параметрів з'єднання, таких як шлях “/api/socket/io”, методи CORS-запитів, інтервали пінгування для перевірки доступності клієнта та таймаути. Далі система починає слухати події підключення нових клієнтів.

Після встановлення з'єднання між сервером та клієнтом, для кожного нового сокет-з'єднання реєструються обробник події newBid (створення нової ставки). Логіка роботи подана на рис. 2.8.

```
io.on('connection', async (socket) => {
  socket.on('newBid', async ({ lotId, amount, userId }) => {
    try {
      const createdBid = await prismaDb?.bid.create({
        data: {
          lotId,
          amount,
          bidderId: userId,
        },
        include: {
          user: {
            select: {
              id: true,
              name: true,
            },
          },
        },
      });

      if (!createdBid) {
        throw new Error('Bid creation failed');
      }

      io.emit('bidCreated', createdBid);
    } catch (error) {
      console.error('BID_CREATION_ERROR ->', error);
      socket.emit('error', 500, 'Failed to create bid');
    }
  });
});
```

Рис. 2.8 – Алгоритм створення нової ставки через WebSocket-з'єднання

При виникненні події newBid сервер приймає дані, що містять ідентифікатор лота, суму ставки та ідентифікатор користувача. Далі здійснюється спроба створення нового запису в базі даних. Паралельно відбувається приєднання до даних ставки основної інформації про користувача (його ім'я та ідентифікатор) для подальшого використання у відповіді.

Якщо під час запису в базу виникає помилка, сервер надсилає повідомлення клієнту, не перериваючи роботу інших клієнтів. У випадку успішного створення ставки сервер формує повідомлення bidCreated, яке транслюється всім підключеним клієнтам для оперативного оновлення інтерфейсів у реальному часі.

Обробка платежів реалізована через обробник POST-запиту, який реагує на події від Stripe. Першим кроком є перевірка достовірності запиту за допомогою підпису stripe-signature. Якщо підпис або секретний ключ відсутні, запит відхиляється з відповідним повідомленням про помилку.

Після успішної валідації підпису із Stripe-події типу `checkout.session.completed` витягуються метадані, передані під час створення сесії платежу: ідентифікатор користувача, ідентифікатори лотів, тип ставки (аукціонна або миттєва), а також дані для доставки (ім'я, прізвище, email, телефон, адреса тощо). Алгоритм обробки події наведено у Додатку А.

Далі система перевіряє наявність усіх обов'язкових полів. У разі їх відсутності – обробка припиняється, а сервер повертає повідомлення про помилку.

У залежності від типу ставки її обробка здійснюється по-різному. У разі, якщо передана ставка була здійснена під час торгів, система оновить виграшну ставку користувача, зазначену в сесії, встановлюючи значення поля `isPaid = true`. Якщо ж йдеться про миттєву покупку, створюється нова ставка з типом `INSTANT`, яка одразу вважається оплаченою та виграшною. Після цього відповідний лот позначається як проданий, тобто поле `isSold` встановлюється у значення `true`.

Наступним кроком система створить запис про доставку. Якщо куплено один лот – створюється один запис, якщо декілька – виконується масове створення записів для кожного лота. У разі отримання події з іншим, непідтримуваним типом, генерується повідомлення про необроблюваний тип події. Успішна обробка завершується відповіддю зі статусом 200.

Висновок до другого розділу

У другому розділі здійснено проєктування платформи для проведення аукціонів, визначено варіанти використання системи відповідно до ролей користувачів та сформовано об'єктно-орієнтовану структуру, яка розмежовує функціонал між звичайними користувачами та адміністраторами. Для наочного відображення взаємодії створено діаграму варіантів використання.

Визначено основні функціональні та нефункціональні вимоги платформи. Розроблено структуру бази даних для зберігання інформації про користувачів, аукціони, лоти, ставки, мультимедійний контент та інші дані, що дозволяє ефективно працювати навіть за високого навантаження.

Описано основні алгоритми роботи системи: створення аукціонів і лотів, проведення ставок у режимі реального часу через WebSocket, обробка платежів через Stripe, автоматичне визначення переможців торгів.

Особливу увагу приділено оновленню ставок у реальному часі, що підвищує зручність і конкурентоспроможність платформи. У підсумку спроектовано архітектуру платформи, яка відповідає сучасним вимогам розробки.

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ ДЛЯ ПРОВЕДЕННЯ АУКЦІОНІВ

3.1 Порядок встановлення та налаштування параметрів системи

Діаграма розгортання [20] на рис. 3.1 ілюструє фізичну архітектуру платформи для аукціонів, демонструючи взаємодію між окремими вузлами та компонентами під час розгортання системи. Вона дозволяє візуально оцінити, як різні частини інфраструктури поєднані між собою та забезпечують стабільну й масштабовану роботу всієї системи.

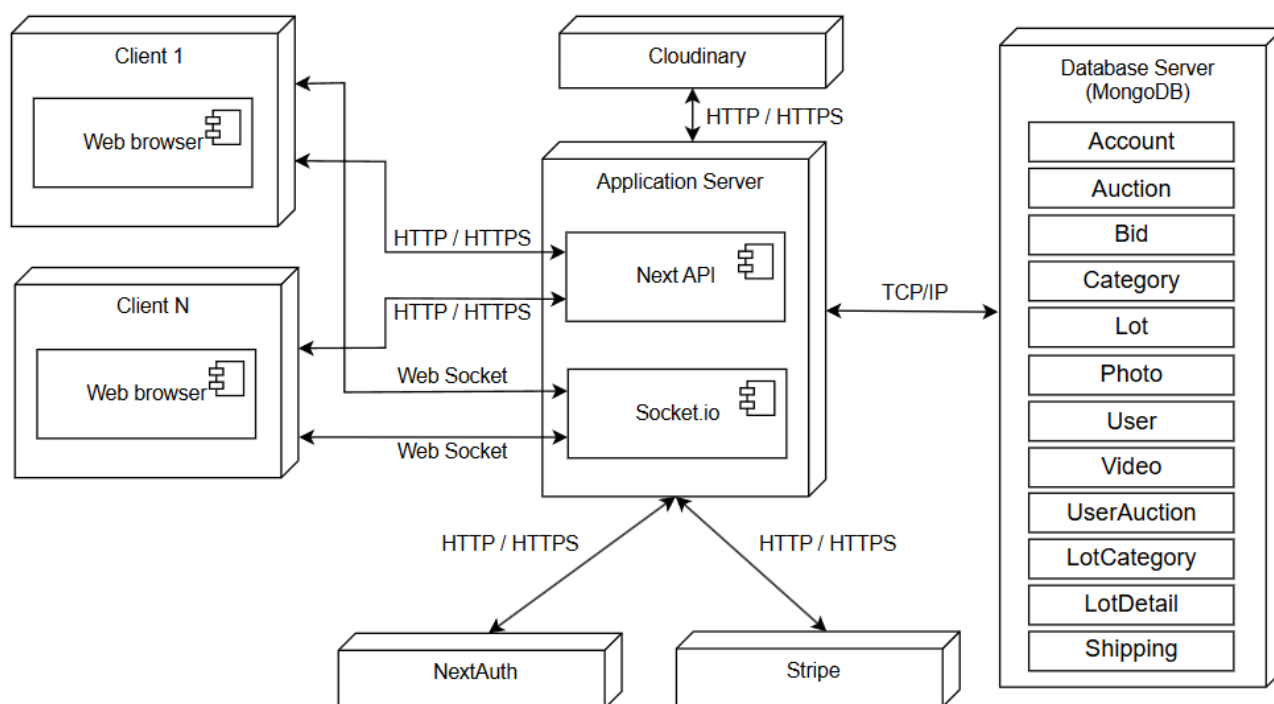


Рис. 3.1 – Діаграма розгортання платформи для проведення аукціонів

Client 1...N – клієнтські пристрої з веббраузерами, через які користувачі отримують доступ до аукціонної платформи. Користувацький інтерфейс завантажується безпосередньо у браузері, а обмін з серверною частиною здійснюється через протоколи HTTP/HTTPS для звичайних запитів і WebSocket – для обробки ставок у реальному часі, що дозволяє отримувати миттєві оновлення та забезпечує безперервну взаємодію.

Application Server – ядро серверної інфраструктури, на якому розгорнуто серверну логіку додатку. Він об'єднує функціональність Next.js API для обробки REST-запитів та використання Socket.io для підтримки WebSocket-з'єднань. Цей

сервер виконує ключові завдання: обробку запитів клієнтів, керування аукціонами, розрахунок ставок та передачу подій у реальному часі.

Database Server (MongoDB) – окремий сервер, відповідальний за зберігання усіх даних системи: інформації про користувачів, лоти, ставки та інші дані. Обмін даними між додатком та базою відбувається по протоколу TCP/IP, що забезпечує надійність і швидкість взаємодії.

Cloudinary – зовнішній сервіс для зберігання й оптимізованої доставки медіа-контенту (зображень та відео лотів). Сервер додатку надсилає запити до Cloudinary через HTTP/HTTPS, використовуючи його API для завантаження, зберігання, масштабування та подальшого використання медіа у фронтенді.

Stripe – інтегрована платіжна система, що відповідає за безпечну обробку фінансових транзакцій. Комунікація з нею також здійснюється через захищені HTTP/HTTPS-запити. Stripe дозволяє користувачам оплачувати виграні лоти.

NextAuth – інтегрований модуль автентифікації користувачів, що відповідає за безпечний вхід до системи. Комунікація з ним здійснюється через HTTP/HTTPS-запити. NextAuth дозволяє користувачам входити в систему за допомогою зовнішніх провайдерів та облікових даних.

Для забезпечення взаємодії між усіма компонентами системи використовуються різні комунікаційні протоколи, що відповідають за передачу даних, синхронізацію в реальному часі та стабільне з'єднання.

Комунікаційні протоколи:

- **HTTP/HTTPS** – використовується для стандартного обміну даними між клієнтами, сервером, Stripe, Cloudinary та NextAuth;
- **WebSocket** – забезпечує двосторонній зв'язок у реальному часі для обробки ставок;
- **TCP/IP** – використовується для стабільного та швидкого з'єднання між сервером додатку та базою даних.

Обрана архітектурна модель дозволяє системі бути гнучкою, адаптивною до навантажень та надає можливість масштабування в разі зростання кількості користувачів. Інтеграція зі сторонніми сервісами, такими як Cloudinary та Stripe,

підвищує ефективність системи та оптимізує витрати на підтримку інфраструктури.

3.2 Структура інтерфейсу платформи для проведення аукціону

Взаємодія з платформою починається зі сторінки автентифікації, яка є першим кроком для доступу до функціональності аукціону (рис. 3.2).

The screenshot shows the Bidium login interface. At the top, there is a navigation bar with the Bidium logo, a language selector set to 'UA', a 'Вхід' (Login) link, and a red 'Реєстрація' (Registration) button. The main area features a white login card with the title 'Увійти' (Login) and the instruction 'Введіть свою електронну пошту та пароль, щоб увійти' (Enter your email and password to log in). There are two input fields: one for the email (containing 'example@gmail.com') and one for the password (with a toggle icon). A prominent red 'Увійти' button is positioned below the password field. Underneath this button is a link 'АБО ПРОДОВЖИТИ ЧЕРЕЗ' (OR CONTINUE WITH), followed by buttons for logging in via GitHub and Google. At the bottom of the card, a link asks 'Ще не маєте облікового запису? Зареєструватися' (Don't have an account? Register). The footer of the page includes the copyright notice '© 2025 Bidium. Всі права захищені.' and social media icons for Facebook, Twitter, and Instagram.

Рис. 3.2 – Сторінка автентифікації

Сторінка автентифікації призначена для входу зареєстрованих користувачів на платформу. Вона містить форму, де потрібно вказати електронну пошту та пароль. Для зручності також доступний вхід через сторонні сервіси, такі як Google та GitHub, що дозволяє швидко отримати доступ без створення нового пароля.

У верхній частині розміщене головне меню навігації з посиланнями на ключові розділи платформи, перемикач мови та кнопки для входу або реєстрації. Нижня частина містить футер із посиланнями на політику конфіденційності, умови використання та соціальні мережі.

Сторінка реєстрації представлена на рис. 3.3. Форма реєстрації містить лише наступні поля: електронну пошту, пароль та підтвердження пароля.

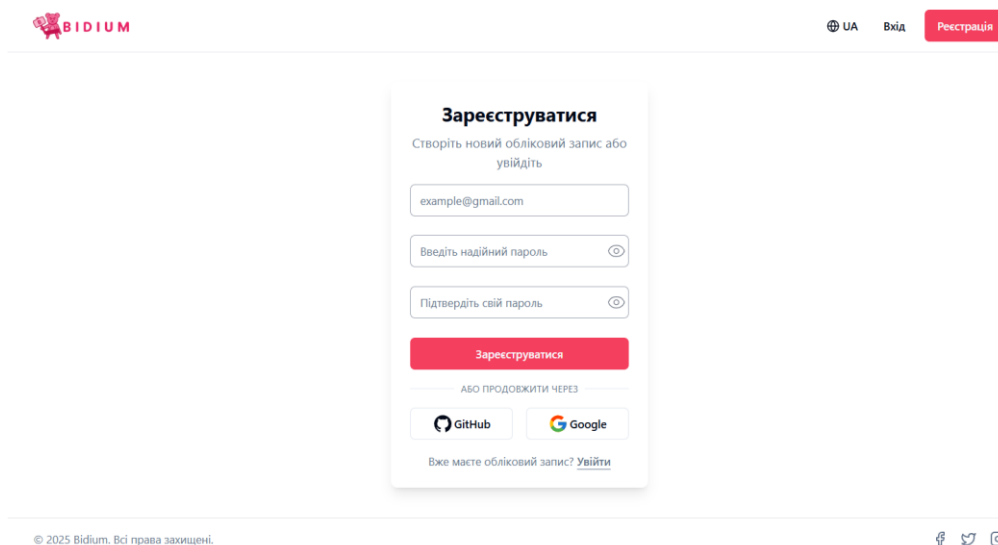


Рис. 3.3 – Сторінка реєстрації

Для забезпечення коректності введених даних реалізована валідація: перевіряється правильність формату електронної пошти, а також співпадіння пароля з його підтвердженням.

Після успішного заповнення форми та її відправки, користувач отримує лист на пошту для підтвердження реєстрації (рис. 3.4).

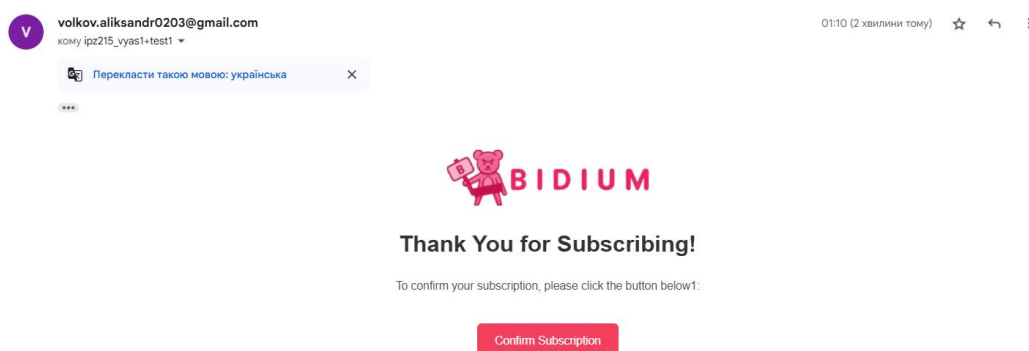


Рис. 3.4 – Лист для підтвердження реєстрації

Після авторизації користувач потрапляє на головну сторінку платформи (рис. 3.5) – для звичайного користувача, (рис. 3.6) – для адміністратора, де відображаються всі активні та завершені аукціони. Ліворуч розташована вертикальна панель навігації, яка дозволяє переходити між головними розділами платформи такими як “Головна”, “Усі аукціони”, “Замовлення” та “Налаштування” для звичайного користувача та додатково “Огляд”, “Користувачі” для адміністратора. Центральна частина інтерфейсу відведена під відображення карток аукціонів, де кожна картка містить назву, кількість лотів,

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						45

діапазон цін, інформацію про категорії, час до завершення аукціону або його статус, а також кнопку для переходу до перегляду лотів.

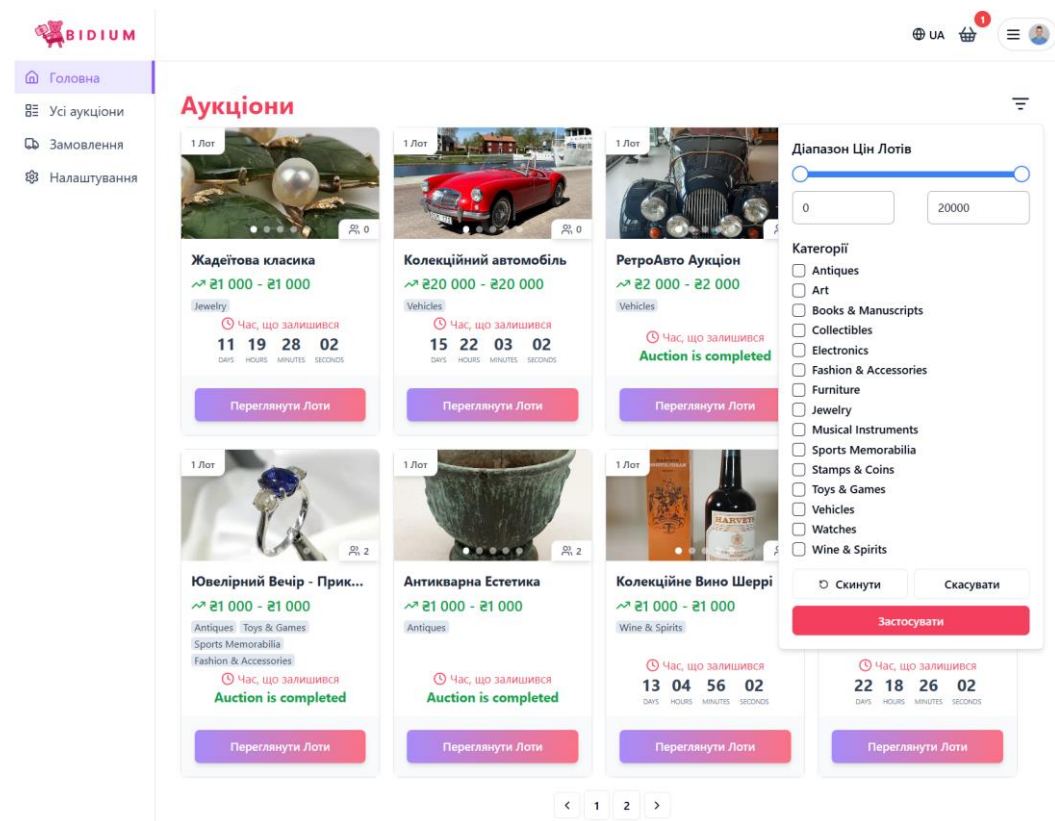


Рис. 3.5 – Головна сторінка платформи для звичайного користувача

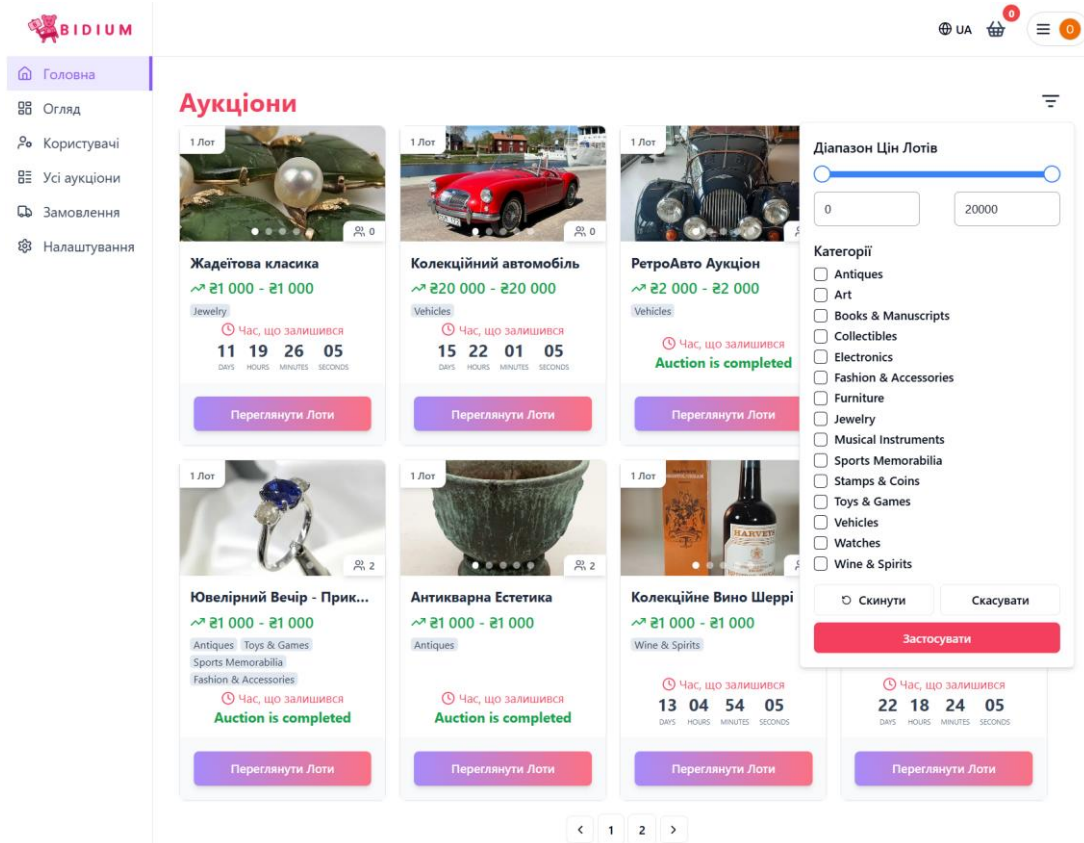
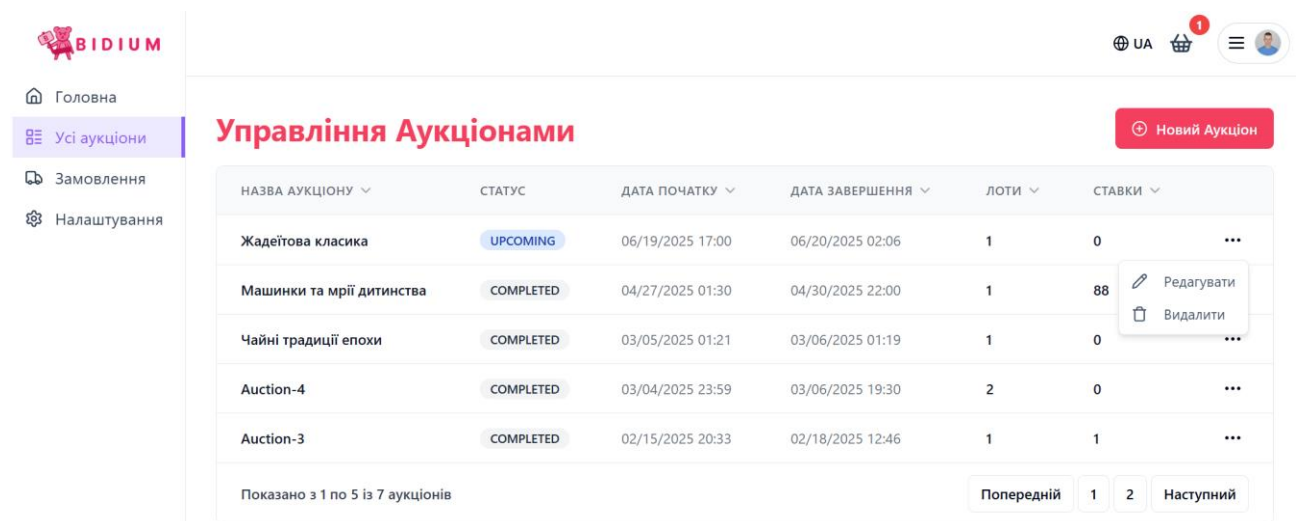


Рис. 3.6 – Головна сторінка платформи для адміністратора

Праворуч розміщується панель фільтрації, яка дозволяє користувачеві обрати бажаний діапазон цін та категорії товарів, після чого застосувати або скинути фільтри. У верхній частині інтерфейсу передбачено елементи керування мовою інтерфейсу, переглядом кошика та доступом до особистого кабінету користувача.

Для переходу на сторінку створення акціону користувачу спочатку потрібно натиснути на кнопку “Усі аукціони”. Перед ним відкриється таблиця з усіма аукціонами, які були створені ним (рис. 3.7).



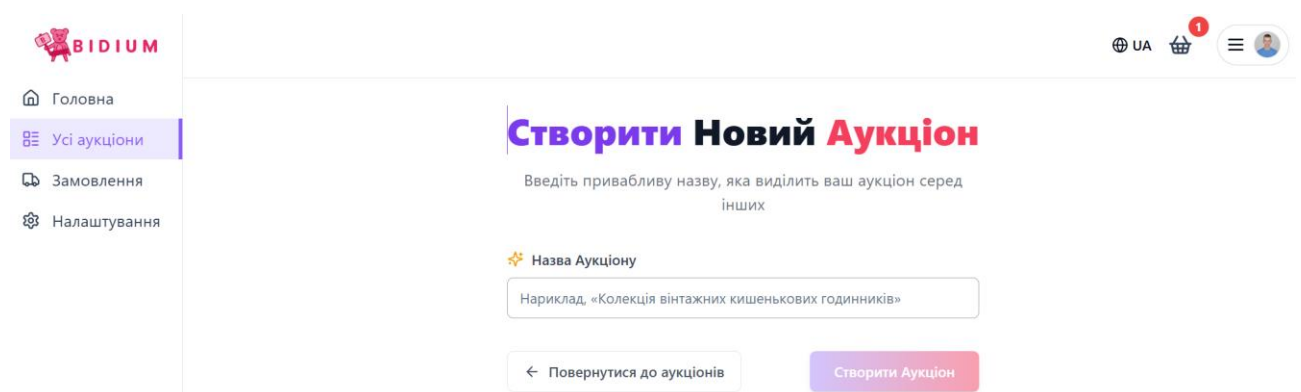
НАЗВА АУКЦІОНУ	СТАТУС	ДАТА ПОЧАТКУ	ДАТА ЗАВЕРШЕННЯ	ЛОТИ	СТАВКИ	
Жадеїтова класика	UPCOMING	06/19/2025 17:00	06/20/2025 02:06	1	0	...
Машинки та мрії дитинства	COMPLETED	04/27/2025 01:30	04/30/2025 22:00	1	88	Редагувати Видалити
Чайні традиції епохи	COMPLETED	03/05/2025 01:21	03/06/2025 01:19	1	0	...
Auction-4	COMPLETED	03/04/2025 23:59	03/06/2025 19:30	2	0	...
Auction-3	COMPLETED	02/15/2025 20:33	02/18/2025 12:46	1	1	...

Показано з 1 по 5 із 7 аукціонів

Попередній 1 2 Наступний

Рис. 3.7 – Сторінка з переліком створених аукціонів даним користувачем

На даній сторінці є можливість перейти у вікно створення аукціону, редагування або ж здійснити його видалення. Після натискання на кнопку “Новий аукціон” користувач буде перенаправлений на сторінку створення аукціону (рис. 3.8).



Створити Новий Аукціон

Введіть привабливу назву, яка виділить ваш аукціон серед інших

Назва Аукціону

Наприклад, «Колекція вінтажних кишенькових годинників»

← Повернутися до аукціонів Створити Аукціон

Рис. 3.8 – Сторінка створення аукціону

На даній сторінці користувачу слід ввести назву аукціону і натиснути кнопку “Створити аукціон”. Після успішного створення аукціону користувач буде перенаправлений на сторінку налаштувань аукціону (рис. 3.9).

BIDUМ

UA

Головна

Усі аукціони

Замовлення

Налаштування

Настрої Аукціону

Опублікувати

Назва*

Антикварна китайська ваза

Дата початку*

6/7/2025, 11:40:00 PM

Опис

Ексклюзивний аукціон рідкісних антикварних предметів з Азії. У-

Дата завершення*

6/7/2025, 11:55:00 PM

Лоти аукціону*

+

Китайська ваза з драконами

Рис. 3.9 – Сторінка налаштувань аукціону

На даній сторінці користувач може змінити назву, додати опис, лоти, дату початку та завершення, а також здійснити публікацію, після заповнення всіх необхідних полів. Адміністратор може погодити даний лот, якщо він відповідає політиці платформи, натиснувши кнопку “Підтвердити”.

Після натискання на створений лот, користувач буде перенаправлений на сторінку налаштувань лоту (рис. 3.10).

Головна

Усі аукціони

Замовлення

Налаштування

Налаштування лоту

Назва*

Китайська ваза з драконами

Опис

Розкішна порцелянова ваза у стилі famille rose з баратим ручним

Категорії*

Antiques Furniture

Деталі (з використанням ШІ для вибору іконок)

Матеріал: порцеляна

Походження: Індія

Початкова ставка*

1500

Мінімальний крок ставки*

50

Ціна миттєвої покупки

4000

Аукціон > Лот

🗑

🖼

Перетягніть сюди зображення, або натисніть для вибору зображення

🖼

🖼

🖼

🖼

🖼

Завантажити зображення 📁

🎥

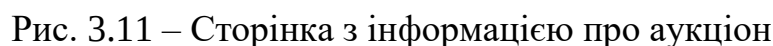
Перетягніть сюди відео, або натисніть для вибору відео

🎥

Завантажити відео 📁

Рис. 3.10 – Сторінка налаштувань лоту

Натиснувши на карточку аукціону, розміщену на головній сторінці сайту, користувач переходить до сторінки з детальною інформацією про аукціон та створені для нього лоти (рис. 3.11). У верхній частині сторінки відображається загальна назва аукціону, його дата початку та завершення, а також короткий опис. Нижче розміщується картка самого лота з фотографіями та відео, описом, стартовою ціною, поточною ставкою, ціною викупу та категорією товару. Також вказано час, що залишився до початку аукціону, і статус лота.



UA

[Головна](#)
[Усі аукціони](#)
[Замовлення](#)
[Налаштування](#)

Лот Аукціону

Китайська ваза з драконами

Lot: 6844a177bc8a6b9d68d23aca

Дата Початку: 07.06.2025, 23:40:00

Дата Завершення: 07.06.2025, 23:55:00

час до початку
00 00 00 00
дні години хвилини секунди

Завершено

Огляд Деталі Процес Ставок Купити Зараз

Інформація Про Ставки

Початкова Ставка	Ціна Купівлі Зараз	Мінімальний Приріст
€1500	€4000	€50

Категорії

Antiques Furniture

Опис

Розкішна порцелянова ваза у стилі famille rose з багатим ручним розписом та декоративними елементами у вигляді золочених ручок-слонів. Основна тематика – дракони серед хмар, що символізують силу й процвітання. Імовірно, період пізньої династії Цін. Відмінний стан, збережено яскраві кольори та орнамент.

Рис. 3.12 – Сторінка лоту (вкладка огляд)

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						49

На даній сторінці розміщено слайдер, де можна переглянути зображення лоту та відео. Поруч зображено блок із вкладками, де розміщено загальний огляд, деталі, процес ставок та можливість купити товар миттєво. У секції огляду зазначена стартова ціна, ціна миттєвого викупу, а також мінімальний крок ставки. Крім того, вказані дата початку і завершення аукціону, категорія товару та його унікальний ідентифікатор. У нижній частині міститься детальний опис. Така структура дозволяє користувачеві швидко ознайомитись з усіма важливими параметрами лоту й прийняти рішення щодо участі в торгах або миттєвої покупки.

На вкладці “Деталі” (рис. 3.13) користувач має змогу переглянути детальний опис лоту, якщо такий був заданий при створенні лоту.

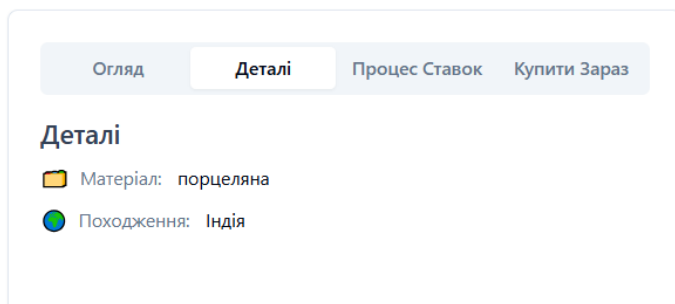


Рис. 3.13 – Вкладка “Деталі”

На вкладці “Процес ставок” (рис. 3.14) користувач має змогу зробити ставку для лоту, якщо торги для даного лоту тривають в даний момент або переглянути історію ставок, якщо процес торгів був завершений.

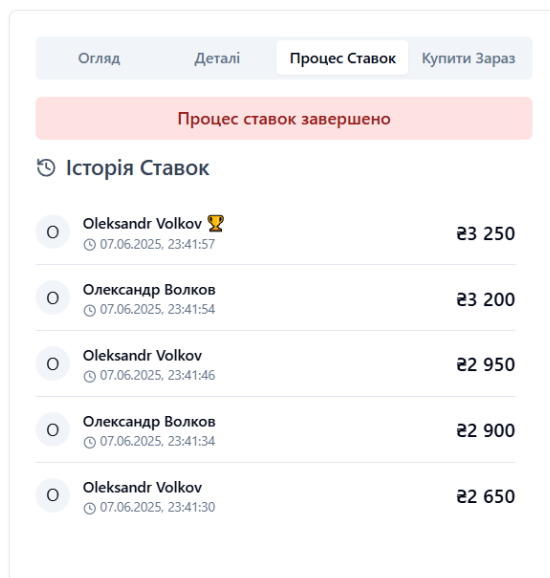


Рис. 3.14 – Вкладка “Процес Ставок” для завершених торгів

На вкладці “Купити Зараз” (рис. 3.15) користувач має змогу придбати лот ще до початку торгів, за умови, що для лоту було встановлено ціну миттєвого викупу при створенні.

Огляд
Деталі
Процес Ставок
Купити Зараз

Ціна моментального викупу: **\$30 000**

Моментальний викуп дозволяє вам придбати лот за фіксованою ціною без участі в аукціоні.

Переваги моментального викупу:

- Гарантоване отримання лоту
- Відсутність конкуренції з іншими учасниками
- Швидке завершення угоди

Купити зараз за \$30 000

Натискаючи кнопку "Купити зараз", ви погоджуєтесь з умовами покупки та правилами сервісу.

Рис. 3.15 – Вкладка “Купити Зараз”

Під час процесу ставок (рис. 3.16) користувачі можуть робити свої пропозиції вручну або скористатися функцією автоставки. Кожен учасник може вказати суму, яка перевищує поточну, з урахуванням встановленого кроку ставки. Усі ставки відображаються в історії з іменем учасника та часом. Аукціон триває до вказаної дати завершення, після чого переможцем стає той, хто зробив найвищу ставку.

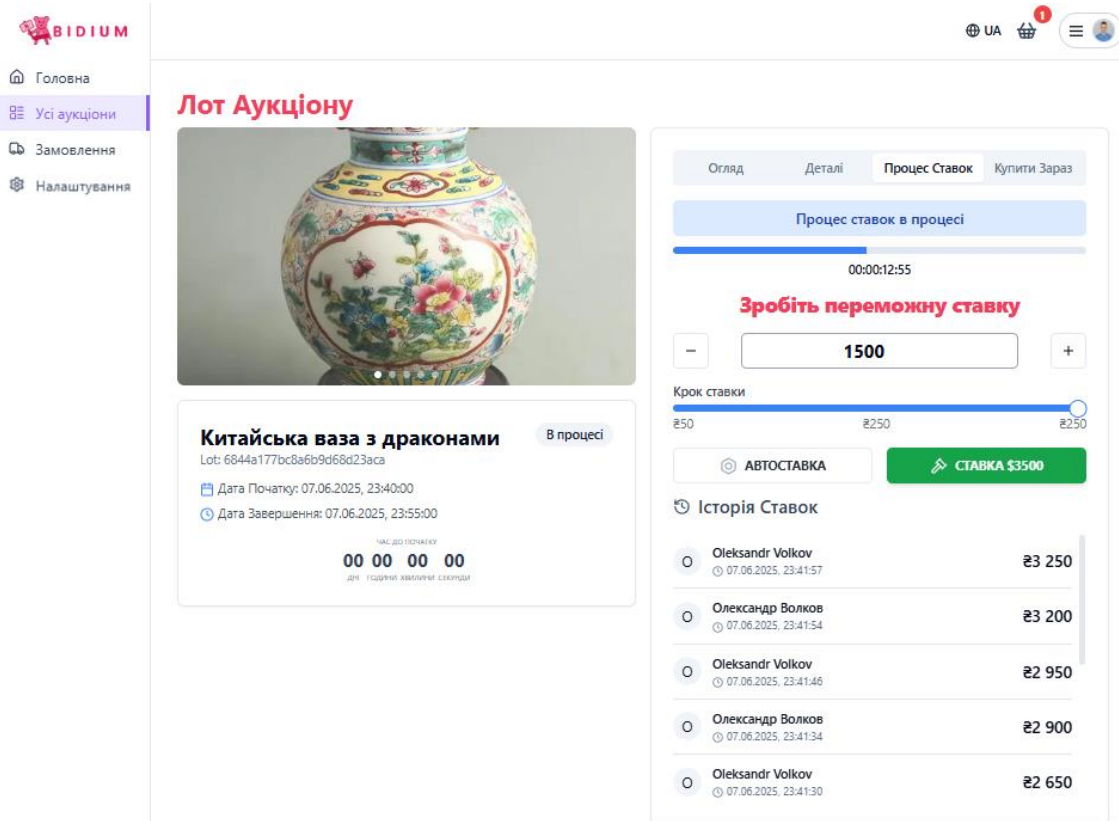


Рис. 3.16 – Вікно з процесом ставок

Після виграшу лоту користувач може побачити його на сторінці “корзина” (рис. 3.17)

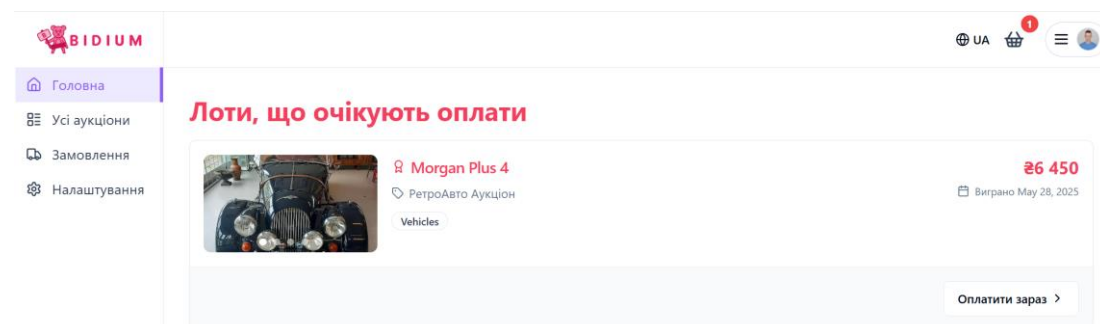


Рис. 3.17 – Сторінка з виграними лотами

Користувач може оплатити даний лот, натиснувши кнопку “Оплатити зараз”, попередньо пройшовши три кроки оформлення замовлення (рис. 3.18 - 3.20) відповідно. На першому етапі потрібно ввести особисту інформацію, таку як ім’я, прізвище, номер телефону та електронну пошту. Далі користувач переходить до другого кроку, де необхідно зазначити адресу доставки, включаючи місто, поштовий індекс і точну адресу. Після заповнення всіх полів відкривається фінальний етап, на якому можна перевірити введені дані та здійснити оплату, натиснувши відповідну кнопку.

Оформлення замовлення

⌚ Час завершення: ~2 хв

1

2

3

👤 Особиста інформація

Ім'я

Олександр

Прізвище

Волков

Телефон

+380 96 052 77 77

Електронна пошта

ipz215_vyas1@student.ztu.edu.ua

Наступний →

Рис. 3.18 – Заповнення особистої інформації

Оформлення замовлення

⌚ Час завершення: ~2 хв

✓

2

3

📄 Адреса доставки

Місто

Житомир

Поштовий індекс

10020

Адреса

Велика Бердичівська 3

← Назад

Наступний →

Рис. 3.19 – Заповнення інформації про адресу доставки

Оформлення замовлення

⌚ Час завершення: ~2 хв

✓

✓

3

📄 Підтвердження замовлення

Особиста інформація:

Олександр Волков
+380 96 052 77 77
ipz215_vyas1@student.ztu.edu.ua

Адреса доставки:

Житомир, 10020
Велика Бердичівська 3

Загальна сума:

\$6450.00

← Назад

Оплатити 📄

Рис. 3.20 – Сторінка з підтвердженням замовлення

Після натискання кнопки “Оплатити” користувач буде перенаправлений на сторінку оплати сервісу Stripe.

На сторінці “Замовлення” (рис. 3.21) користувач має змогу переглянути всі замовлення, свої замовлення – ті лоти, які він виграв, та надіслані замовлення – лоти, які були ним створено та які він має надіслати переможцю аукціону.

Замовлення

Усі Замовлення Мої Замовлення Надіслані Замовлення

КЛІЄНТ	КОНТАКТ	АДРЕСА	ЛОТ	СТАТУС	ДАТА
Олександр Волков Замовлення #1	ipz215_vyas1@student.ztu.edu.ua +380 96 052 81 19	Dmitra Dontsova 3, Zhytomyr, 10014	Watch	CANCELED	Створено 03/08/2025 17:06 Оновлено 04/30/2025 03:17
Дмитро Волков Замовлення #2	ipz215_vyas1@student.ztu.edu.ua +380 96 052 81 19	Dmitra Dontsova 3, Zhytomyr, 10014	Lot-1	DELIVERED	Створено 02/23/2025 18:28 Оновлено 04/30/2025 02:35

Показано з 1 по 2 із 2 замовлення

Попередній 1 Наступний

Рис. 3.21 – Сторінка з замовленнями

На сторінці з інформацією про доставку (рис. 3.22) користувач має змогу переглянути відповідну інформацію щодо замовлення, статусу доставки, оплати та деталей лоту.

Lot Delivery Information

Статус доставки Скасовано

Замовлення 67cc5cf1413a9445dc01d9aa від 03/08/2025 17:06

Олександр Волков
Клієнт

Місце призначення
Dmitra Dontsova 3
Zhytomyr, 10014

Контактна інформація
+380 96 052 81 19
ipz215_vyas1@student.ztu.edu.ua

Оплата

Статус оплати Оплачено

Сума до сплати **15000 €**

Інформація про лот



Watch
Лот #67cc5a59413a9445dc01d9a2

[Переглянути деталі лота](#)

Рис. 3.22 – Сторінка інформації про доставку

На сторінці “Налаштування” (рис. 3.23) користувачі мають змогу змінити пароль.

На сторінці “Усі користувачі” (рис. 3.25) адміністратор має можливість переглядати всіх зареєстрованих користувачів платформи, редагувати інформацію про них, а також активувати та деактивувати аккаунти.

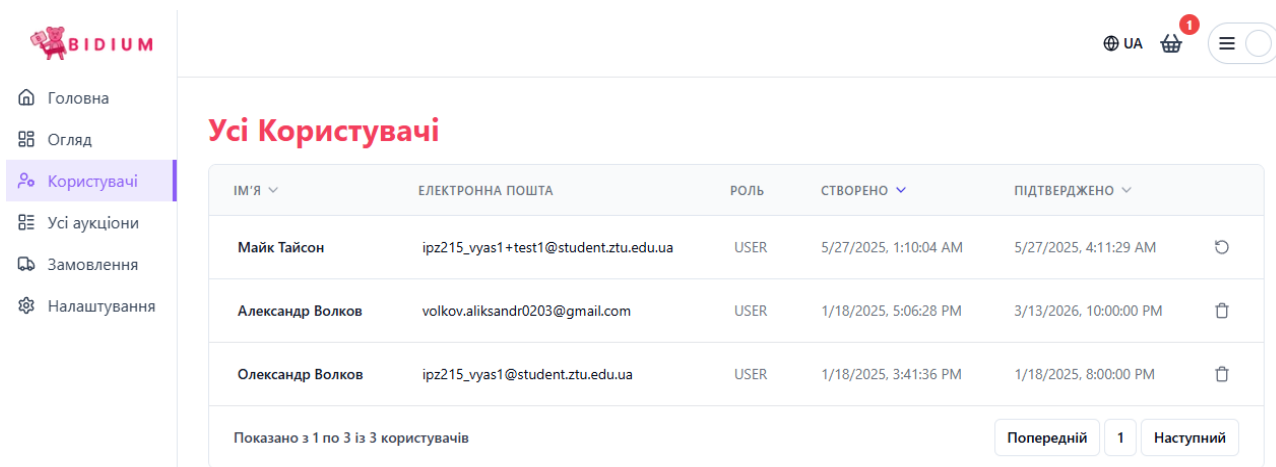


Рис. 3.25– Сторінка з списком користувачів платформи

3.3 Тестування роботи програмного комплексу

Тестування програмного продукту платформи для проведення аукціонів проводилося на завершальному етапі розробки з метою відповідності реалізованої системи вимогам технічного завдання. Основною метою тестування було забезпечити надійність, безпечність і правильність роботи системи, виявити та усунути можливі помилки ще до етапу впровадження.

Особливу увагу було приділено сценаріям, які найбільш часто зустрічаються під час реального використання системи, таким як здійснення ставок, завершення аукціону та визначення переможця, а також інтерактивна взаємодія між користувачами та інтерфейсом у режимі реального часу.

Для ефективної перевірки були застосовані методи ручного та автоматизованого тестування. Усі виявлені дефекти було класифіковано за критичністю, після чого усунуто під час етапу виправлення.

Перед початком тестування було сформовано набір тест-кейсів, що охоплюють основні функціональні можливості платформи для проведення аукціонів. Тест-кейси були розроблені на основі типових сценаріїв взаємодії користувача із системою та критичних бізнес-процесів. Під час розробки значний акцент було зроблено на такі аспекти, як реєстрація та авторизація користувачів, створення та участь в аукціонах, динамічне оновлення ставок у

реальному часі, коректне завершення торгів, а також інтеграція з платіжною системою Stripe. Кожен тест-кейс містить опис вхідних даних, очікуваний результат, послідовність кроків тестування та статус проходження. Нижче наведено приклади ключових тест-кейсів, що були виконані в межах тестування платформи (табл. 3.1).

Таблиця 3.1

Таблиця тестових сценаріїв

№	Назва тест-кейсу	Вхідні дані	Очікуваний результат	Кроки тестування	Статус
1	Реєстрація нового користувача	Email, пароль, підтвердження паролю	Успішна реєстрація	1. Перейти на сторінку реєстрації 2. Ввести коректні дані 3. Натиснути кнопку "Зареєструватися"	Пройдено
2	Авторизація з правильно введеними даними	Існуючі email та пароль	Вхід у систему (корінь сайту)	1. Перейти на сторінку входу 2. Ввести існуючі облікові дані 3. Натиснути "Увійти"	Пройдено
3	Створення нового аукціону	Назва	Аукціон створено і відображається у таблиці всіх аукціонів даного користувача	1. Увійти до системи 2. Перейти до розділу "Мої аукціони" 3. Натиснути "Створити аукціон" 4. Задати назву аукціону 5. Зберегти	Пройдено
4	Ставка в реальному часі	Вища ставка від поточної	Ставка приймається, оновлення суми ставки у всіх клієнтів	1. Відкрити активний аукціон 2. Ввести суму ставки 3. Натиснути "Зробити ставку"	Пройдено
5	Завершення аукціону	Пройдений час на здійснення ставки	Визначення переможця, торги завершено	1. Дочекатися закінчення часу відведеного для здійснення ставки 2. Підтвердити, що подання ставок заблоковано	Пройдено

Продовження таблиці 3.1

6	Авторизація з неправильними даними	Невірний email або пароль	Повідомлення про помилку	1. Перейти на сторінку входу 2. Ввести неіснуючий email або неправильний пароль 3. Натиснути "Увійти"	Пройдено
7	Фільтрація за категорією	Категорія: "Картини"	Виводяться лише лоти з відповідної категорії	1. Вибрати категорію 2. Перевірити, що фільтр спрацював	Пройдено
8	Ініціація платежу через Stripe	Валідні реквізити картки, сума	Переадресація на форму Stripe, успішна оплата	1. Перейти до сторінки оплати лота 2. Натиснути "Оплатити" 3. Система відкриває форму Stripe 4. Ввести дані тестової картки 5. Натиснути "Оплатити"	Пройдено
9	Вихід із системи	Клік на "Вийти"	Користувач перенаправлений на сторінку входу, сесія завершена	1. Увійти в систему 2. Натиснути "Вийти"	Пройдено

Також за допомогою інструменту Lighthouse [21] було проведено детальний аналіз вебсайту, що дозволило оцінити його ключові характеристики за кількома важливими критеріями: продуктивність, доступність, відповідність вебстандартам та SEO-оптимізація. Кожен з цих показників відображає конкретний аспект якості сайту – від швидкості завантаження та взаємодії з користувачем до відповідності сучасним вимогам пошукових систем та підтримки доступності для всіх груп користувачів. Нижче наведено зображення, що демонструють ключові показники Lighthouse.

На рис. 3.26 представлено результати оцінки відповідності сайту сучасним вебстандартам та рівня SEO-оптимізації. Ці показники дають уявлення про технічну якість верстки та ефективність сайту в пошуковому просуванні.

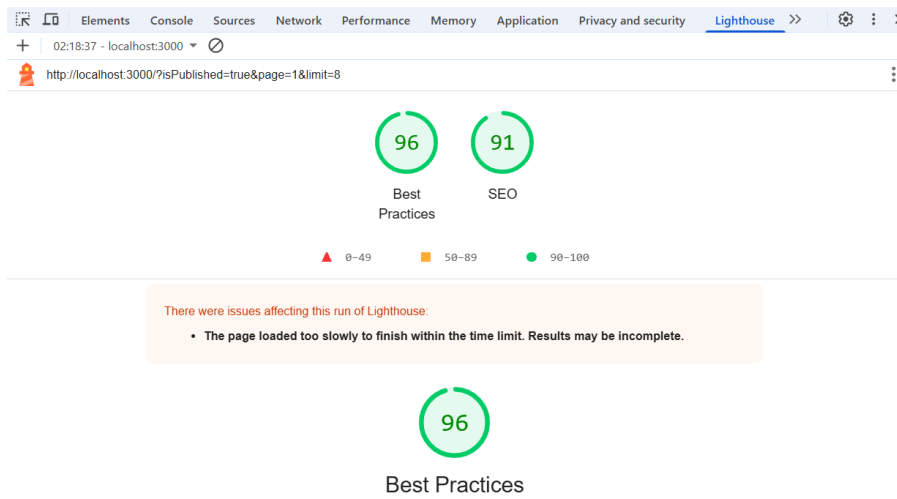


Рис. 3.26 – Розрахунок показників відповідності веб-стандартам та SEO оптимізації

На рис. 3.27 зображено результати аналізу продуктивності та доступності сайту. Ці метрики дозволяють оцінити швидкість завантаження, інтерактивність та зручність користування для різних категорій користувачів.

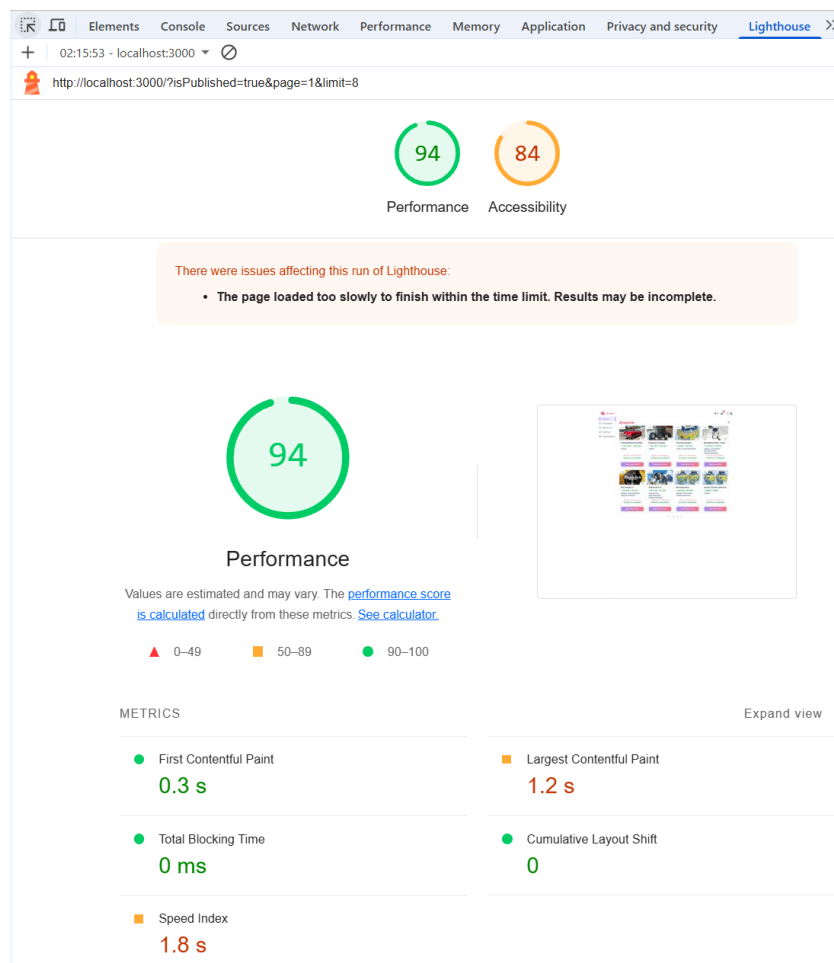


Рис. 3.27 – Розрахунок показників продуктивності та доступності сайту

На основі результатів, представлених на зображеннях, можна зробити висновок, що сайт демонструє високі показники якості. Зокрема, продуктивність оцінена на 94 бали, що свідчить про швидке завантаження контенту. Також добре виглядають “Best Practices” (96 балів) і SEO (91 бал), що вказує на дотримання сучасних стандартів, адаптивності та оптимізації для пошукових систем. Дещо нижчий показник доступності (84 бали) свідчить про наявність незначних недоліків у підтримці користувачів з обмеженими можливостями, які доцільно покращити. Загалом, Lighthouse підтвердив високий рівень технічної реалізації проєкту, хоча подальша оптимізація, особливо в частині доступності, дозволить досягти ще кращих результатів.

Висновок до третього розділу

У третьому розділі було детально розглянуто інтерфейс та порядок роботи платформою для проведення аукціонів. Було описано архітектуру системи, включаючи фізичну структуру серверів і зовнішніх сервісів, що забезпечують її ефективну та масштабовану роботу. Особливу увагу приділено інтерфейсу користувача: від етапу реєстрації й автентифікації до участі в аукціонах, перегляду лотів, здійснення ставок, оформлення замовлень.

Розділ містить ілюстрації ключових сторінок інтерфейсу як для звичайних користувачів, так і для адміністратора, що дозволяє повноцінно оцінити функціональні можливості платформи та її зручність. Також наведено приклади тестових сценаріїв, які підтверджують працездатність основних функцій і відповідність реалізації технічному завданню.

Загалом, платформа характеризується інтуїтивно зрозумілим інтерфейсом, широким функціоналом і стабільною технічною реалізацією, що забезпечує якісне проведення онлайн-аукціонів.

ВИСНОВКИ

У процесі виконання випускної кваліфікаційної роботи було розроблено платформу для проведення аукціонів, що дозволяє автоматизувати ключові етапи торгів – від створення лота до визначення переможця. Робота охопила повний цикл створення програмного продукту: від аналізу предметної області до реалізації інтерфейсу та тестування готової системи.

На етапі дослідження були розглянуті сучасні підходи до реалізації аукціонних платформ, здійснено аналіз існуючих аналогів, визначено переваги та недоліки наявних рішень. Це дозволило обґрунтувати вибір архітектури та інструментальних засобів, які найкраще відповідають поставленим завданням.

Під час проєктування системи було побудовано об'єктно-орієнтовану модель, розроблено структуру бази даних та реалізовано основні алгоритми, що забезпечують роботу механізму аукціонних торгів. Система підтримує створення, редагування та публікацію лотів, подання ставок, автоматичне визначення переможця та завершення торгів.

Окрему увагу приділено створенню зручного та інтуїтивного інтерфейсу користувача, а також забезпеченню налаштування та простоти встановлення програмного комплексу. Проведене тестування підтвердило стабільність роботи, коректність реалізованого функціоналу та відповідність програмного продукту початковим вимогам.

У підсумку, розроблена система демонструє можливість створення платформи для проведення аукціонів із використанням сучасних інформаційних технологій. Реалізована архітектура, основні компоненти функціоналу та інтерфейс користувача забезпечують базову основу для подальшого вдосконалення та розширення можливостей системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Волков О. М., Савіцький Р. С. ВПЛИВ SSR ТА SSG НА ПРОДУКТИВІСТЬ ТА SEO. *ТЕЗИ ДОПОВІДЕЙ Інформаційно-комп'ютерні технології* : XV Міжнародної науково-технічної конференції, Житомир, Україна, 28–29 берез. 2025. С. 21–22. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2025/04/povnyj-tekst.pdf>.
2. Волков О. М., Савіцький Р. С. ВПЛИВ UNIVERSAL SENTENCE ENCODER НА РОЗВИТОК ОБРОБКИ ПРИРОДНОЇ МОВИ. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ –2025 : Збірник тез XII Всеукраїнської науково-практичної конференції молодих учених, Київ, Україна, 15 трав. 2025. Київ, 2025. С. 18–20. URL: <https://zcit.kubg.edu.ua/index.php/journal/issue/view/13/23>.
3. Surplex. URL: <https://www.surplex.com/en>.
4. Prozorro.sale. URL: <https://prozorro.sale/about/>.
5. Ebay. URL: <https://www.ebay.com/>.
6. Одинак. *Refactoring Guru. Породжувальні патерни*. URL: <https://refactoring.guru/uk/design-patterns/singleton>.
7. TypeScript. Utility Types. URL: <https://www.typescriptlang.org/docs/handbook/utility-types.html>.
8. Vercel. *Next.js*. URL: <https://nextjs.org/>.
9. Meta. *React.dev*. URL: <https://react.dev/>.
10. Vercel. *Shadcn*. URL: <https://ui.shadcn.com/>.
11. Tailwind Labs Inc. *Tailwindcss*. URL: <https://tailwindcss.com/docs>.
12. *TanStack. TanStack Query*. URL: <https://tanstack.com/query/latest/docs/framework/react/overview>.
13. *Next Cloudinary*. URL: <https://next.cloudinary.dev>.
14. *NextAuth.js*. URL: <https://next-auth.js.org/>.
15. *Mongodb*. URL: <https://www.mongodb.com/>.
16. *Prisma*. URL: <https://www.prisma.io/>.

- 17.Моралес Д. Що таке діаграма варіантів використання UML: символи, шаблони, інструмент і посібник. *Mindonmap*. 02.03.2023. URL: <https://www.mindonmap.com/uk/blog/what-is-a-uml-use-case-diagram/>.
- 18.Моралес Д. Дізнайтеся все про діаграму активності UML [з методами]. *Mindonmap*. 02.03.2023. URL: <https://www.mindonmap.com/uk/blog/uml-activity-diagram/>.
- 19.Моралес Д. Дізнайтеся про важливі деталі діаграми послідовності UML. *MindOnMap*. 02.03.2023. URL: <https://www.mindonmap.com/uk/blog/what-is-uml-sequence-diagram/>.
- 20.Паркер А. Діаграма розгортання: Підручник з UML із ПРИКЛАДОМ. *Guru99*. 26.09.2024. URL: <https://www.guru99.com/uk/deployment-diagram-uml-example.html>.
- 21.Introduction to Lighthouse. *Chrome for developers*. 06.02.2025. URL: <https://developer.chrome.com/docs/lighthouse/overview>.

ДОДАТКИ

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						64

```

import prismaDb from '@lib/prismadb';
import { BidType } from '@prisma/client';
import { headers } from 'next/headers';
import { NextResponse } from 'next/server';
import { Stripe } from 'stripe';

export async function POST(req: Request) {
  try {
    const body = await (await req.blob()).text();
    const signature = headers().get('stripe-signature') as string;

    if (!signature) {
      return new NextResponse('Missing Stripe signature', { status: 400 });
    }

    if (!process.env.STRIPE_WEBHOOK_SECRET) {
      return new NextResponse('Missing webhook secret', { status: 500 });
    }

    let event;
    try {
      event = Stripe.webhooks.constructEvent(body, signature, process.env.STRIPE_WEBHOOK_SECRET);
    } catch {
      return new NextResponse('Webhook signature verification failed', { status: 400 });
    }

    if (event.type === 'checkout.session.completed') {
      const session = event.data.object as Stripe.Checkout.Session;

      const userId = session?.metadata?.userId;
      const lotsId = session?.metadata?.lotsId ? JSON.parse(session?.metadata?.lotsId) : [];
      const address = session?.metadata?.address;
      const city = session?.metadata?.city;
      const postalCode = session?.metadata?.postalCode;
      const phone = session?.metadata?.phone;
      const firstName = session?.metadata?.firstName;
      const lastName = session?.metadata?.lastName;
      const email = session?.metadata?.email;
      const bidType = session?.metadata?.bidType as BidType;

      if (
        !userId ||
        !lotsId ||
        !address ||
        !city ||
        !postalCode ||
        !phone ||
        !firstName ||
        !lastName ||
        !email ||
        !bidType
      ) {
        console.error('Missing required metadata:', {
          userId: !!userId,
          lotsId: !!lotsId,
          address: !!address,
          city: !!city,
          postalCode: !!postalCode,
          phone: !!phone,
        });
      }
    }
  }
}

```

```

        firstName: !!firstName,
        lastName: !!lastName,
        email: !!email,
        bidType: !!bidType,
    });
    return new NextResponse('Webhook Error: Missing required metadata', {
        status: 400,
    });
}

if (bidType === BidType.BIDDING) {
    await prismaDb.bid.updateMany({
        data: {
            isPaid: true,
        },
        where: {
            lotId: {
                in: Array.isArray(lotsId) ? lotsId : [lotsId],
            },
            bidderId: userId,
            isWinner: true,
        },
    });
}
else if (bidType === BidType.INSTANT) {
    const lotId = Array.isArray(lotsId) ? lotsId[0] : lotsId;
    const lot = await prismaDb.lot.findFirst({
        where: {
            id: lotId,
        },
    });
    if (!lot?.buyNowBid) {
        return new NextResponse('Webhook Error: Missing buy now bid', {
            status: 400 });
    }

    await prismaDb.bid.create({
        data: {
            amount: lot.buyNowBid,
            lotId,
            bidderId: userId,
            isPaid: true,
            isWinner: true,
            type: BidType.INSTANT,
        },
    });

    await prismaDb.lot.update({
        data: {
            isSold: true,
        },
        where: {
            id: lotId,
        },
    });
}
}

```

```

const shippingData = {
  email,
  firstName,
  lastName,
  phone,
  city,
  address,
  postalCode,
  userId,
};

if (!Array.isArray(lotsId)) {
  await prismaDb.shipping.create({
    data: {
      ...shippingData,
      lotId: lotsId,
    },
  });
} else {
  await prismaDb.shipping.createMany({
    data: lotsId.map((lotId) => ({
      ...shippingData,
      lotId,
    })),
  });
} else {
  console.error('Unhandled event type: ${event.type}');
}

return new NextResponse(null, { status: 200 });
} catch (error) {
  if (error instanceof Error) {
    return new NextResponse('Webhook Error: ${error.message}', { status: 500
  });
  }

  return new NextResponse('Webhook Error: Unknown error occurred', { status: 500
});
}

```